Міністерство освіти і науки України

ІНФОРМАЦІЙНІ МЕРЕЖІ ТА АДМІНІСТРУВАННЯ

Конспект лекцій
для здобувачів першого (бакалаврського) рівня вищої освіти
освітньої програми
«Інформаційні системи та технології охорони і безпеки»
галузь знані 12/F Інформаційні технології
спеціальності 126/F6 Інформаційні системи та технології
денної та заочної форм навчання

Луцьк 2026

УДК 004.65(07)
І-74

Рекомендовано до видання вченою радою факультету КІТ ЛНТУ,
протокол № _____ від « _____ » _____ 20 26 _____ року.

Голова вченої ради факультету КІТ _____ Інна КОНДІУС

Електронна копія друкованого видання передана для внесення в репозитарій ЛНТУ

Директор бібліотеки _____ Наталія ПОЛІЩУК

Розглянуто і схвалено на засіданні кафедри комп'ютерної інженерії та безпеки
ЛНТУ, протокол № _____ від « ____ » _____ 20 26 року.

Завідувач кафедри КІБ _____ Тарас ТЕРЛЕЦЬКИЙ

Укладач: _____ Наталія БАГНЮК, кандидат технічних наук,
доцент кафедри комп'ютерної інженерії та безпеки ЛНТУ
_____ Олег КАЙДИК, кандидат технічних наук,
доцент кафедри комп'ютерної інженерії та безпеки ЛНТУ

Рецензент: _____ Роман ГРУДЕЦЬКИЙ, старший викладач
кафедри автоматизації та комп'ютерно-інтегрованих технологій,
проректор з НПР та цифрової трансформації ЛНТУ

Відповідальний за випуск: _____ Тарас ТЕРЛЕЦЬКИЙ, кандидат
технічних наук, доцент кафедри комп'ютерної інженерії та безпеки ЛНТУ
Національного університету харчових технологій»

І-74 Інформаційні мережі та адміністрування: конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Інформаційні системи та технології охорони і безпеки» галузі знань 12/F Інформаційні технології спеціальності 126/F6 Інформаційні системи та технології денної та заочної форм навчання / уклад. Н. В. Багнюк, О. Л. Кайдик. Луцьк: ЛНТУ, 2026. 236 с.

Конспект лекцій з дисципліни «Інформаційні мережі та адміністрування» складено відповідно до діючої програми курсу.

Призначене для здобувачів вищої освіти спеціальності 126/F6 Інформаційні системи та технології освітньої програми «Інформаційні системи та технології охорони і безпеки».

ЗМІСТ

Тема 1 Вступ до мереж 5
Тема 2. Адресація в мережах 13
Тема 3. Моделі та протоколи 28
Тема 4. Принципи комутації та маршрутизації 37
Тема 5. Протоколи канального, мережевого та транспортного рівнів 86
Тема 6. Віртуальні локальні мережі та віртуальні приватні мережі 89
Тема 7. Поняття мережної безпеки. Принципи роботи ACL 96
Тема 8. Бездротові мережі 105
Тема 9. Глобальні мережі 108
Тема 10. Віртуалізація та автоматизація роботи мережі 110
Тема 11 Віртуалізація та серверні середовища. Встановлення та основи адміністрування Windows Server 112
Тема 12 Active Directory 153
Тема 13 Групові політики 178
Тема 14 Служба DNS, DHCP 190
Тема 15 Основи Linux-адміністрування. Мережеві служби Linux. Веб-сервісна інфраструктура у Linux 203
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 233

ВСТУП

Конспект лекцій з дисципліни «Інформаційні мережі та адміністрування» розроблено відповідно до чинної освітньої програми та призначено для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 126/F «Інформаційні системи та технології». Матеріал структуровано за тематичними модулями, що дозволяє послідовно опанувати ключові поняття мережевих технологій і сформувати практичні компетентності, необхідні для подальшої професійної діяльності у сфері адміністрування комп'ютерних систем і мереж.

Сучасний етап розвитку інформаційного суспільства характеризується стрімким зростанням обсягів даних, широким упровадженням хмарних технологій, цифровізацією бізнес-процесів та підвищенням вимог до надійності й безпеки інформаційних систем. Комп'ютерні мережі стали основою функціонування практично всіх сфер діяльності − від освіти й науки до промисловості, логістики та критично важливої інфраструктури.

Дисципліна «Інформаційні мережі та адміністрування» є фундаментальною складовою професійної підготовки фахівців у галузі інформаційних технологій. Вона формує системне розуміння принципів побудови комп'ютерних мереж, методів адресації, маршрутизації, комутації, а також засад адміністрування мережевих і серверних середовищ.

Метою вивчення курсу є набуття здобувачами вищої освіти теоретичних знань і практичних навичок проєктування, розгортання, налаштування та супроводу мережевої інфраструктури, забезпечення її стабільної роботи й інформаційної безпеки. Особлива увага приділяється сучасним технологіям Ethernet, VLAN, VPN, бездротовим мережам, службам DNS і DHCP, віртуалізації, основам адміністрування Windows Server і Linux, а також принципам роботи Active Directory та групових політик.

Тема 1 Вступ до мереж

Основні поняття та характеристики мереж. Етапи розвитку комп'ютерних та телекомунікаційних мереж. Класифікація комп'ютерних мереж. Загальні принципи побудови комп'ютерних мереж. Мережеві топології та їх характеристики. Поняття фізичної та логічної топології. Локальні мережі. Основні компоненти локальної мережі. Канали і лінії зв'язку. Кабельні системи. Характеристики ліній зв'язку. Мережеві пристрої.

Однією із суттєвих причин, які прискорили появу комп'ютерів, була потреба в розв'язуванні дуже широкого спектра задач. Між комп'ютерами, які розв'язували схожі завдання, досить часто виникали проблеми обміну даними. Як наслідок, з'явилася ідея об'єднати обчислювальні ресурси різних комп'ютерів, тобто ідея створення комп'ютерної мережі [1].

Комп'ютерна мережа – це сукупність пристроїв, з'єднаних каналами передавання даних, для спільного користування апаратними, програмними та інформаційними ресурсами під керуванням спеціального програмного забезпечення.

Вузол мережі (англ. Node) – це пристрій, з'єднаний з іншими пристроями через мережу. Вузлами можуть бути комп'ютери, мобільні телефони, кишенькові комп'ютери та спеціальні мережні пристрої.

Призначенням комп'ютерних мереж є забезпечення [1]:

– швидкого обміну даними між окремими комп'ютерами мережі;

– спільного використання комп'ютерних програм і даних;

– спільної роботи користувачів над проєктами;

– віддаленого керування комп'ютерами;

– спільного доступу до периферійних пристроїв (принтерів, сканерів, зовнішньої пам'яті);

– спільного доступу до інформаційних ресурсів.

У комп'ютерній мережі комп'ютери можуть виконувати різні функції.

Комп'ютер, який керує розподілом ресурсів мережі, називають сервером (від англ. server − той, хто подає). Комп'ютери, які користуються ресурсами мережі, називають клієнтами або робочими станціями.

Залежно від завдань, які виконують комп'ютери, мережі розрізняють за територією, типом операційної системи, розподілом функцій, інфраструктурою та місцем розташування технічних засобів, які входять у мережу, та ін.

У 1961 році американський інженер українського походження Леонард Клейнрок запропонував ідею пакетної комутації, яка наразі є основою передавання даних мережею. А в 1964 році виклав основні принципи та розробив теорію.

Американського вченого Джозефа К. Р. Ліклайдера часто називають духовним батьком Інтернету. У 1962 році в низці статей він виклав свою концепцію «Галактичної мережі» − прообраз сучасного Інтернету.

Схему класифікації комп'ютерних мереж за різними ознаками наведено на рисунку 1.1.

Розглянемо класифікацію комп'ютерних мереж детально [1].

За територією мережі поділяються таким чином:

– персональні (PAN, від англ. Personal Area Network − мережа особистого простору, персональна мережа) − мережі для взаємодії пристроїв, що належать одній людині та об'єднують її власні електронні пристрої: персональні комп'ютери, ноутбуки, планшети, смартфони, комунікатори;

– локальні (LAN, від англ. Local Area Network − мережа локального простору) − з'єднують пристрої, розташовані на порівняно невеликій відстані один від одного, зазвичай у межах однієї або кількох сусідніх будівель, наприклад мережа навчального закладу;

– міські, регіональні (MAN, від англ. Metropolitan Area Network − мережа міського простору) − обласні й національні мережі;

– глобальні (WAN, від англ. Wide Area Network − мережа широкого простору) −

об’єднують комп’ютерні мережі. Найвідомішою глобальною мережею є Інтернет [1].

Рисунок 1.1 – Класифікація мереж [1]

Сучасні операційні системи (ОС) поділяються на спеціалізовані та мережеві.

Спеціалізовані ОС призначені для роботи з мережевим обладнанням певної компанії. Так, Cisco IOS (англ. Internetwork Operating System − міжмережева ОС) працює виключно з маршрутизаторами й комутаторами компанії Cisco, а ZyNOS − ОС компанії ZyXEL, працює з маршрутизаторами Prestige [1].

За розподілом функцій між комп’ютерами мережі поділяють на однорангові й клієнт-серверні.

2. Мережеві топології

Комп’ютерні мережі поділяються також за топологією.

Мережна топологія визначає структуру мережі. В топології мережі можна виділити дві складові. Перша − це фізична топологія, яка визначає шлях прокладки кабелю та середовище передачі. Друга – це логічна топологія, що визначає, яким чином хости мають доступ до середовища передачі даних.

Існують три базові топології («загальна шина», «кільце», «зірка») та додаткові, що є модифікацією або поєднанням базових, наприклад топологію «дерево» можна розглядати як комбінацію декількох «зірок».

Кожна топологія накладає певні вимоги [1].

Топологія «загальна шина» передбачає використання одного кабелю, до якого під’єднуються всі комп’ютери мережі (рис. 1.2). Надіслане з будь-якого комп’ютера мережі повідомлення поширюється на всі інші комп’ютери мережі. Кожний із них перевіряє, кому адресовано повідомлення. Опрацьовує повідомлення лише той комп’ютер, якому воно адресоване. Комп’ютери можуть передавати дані лише послідовно, оскільки лінія зв’язку одна і спільна. Всі комп’ютери мають рівні права, все обладнання є ідентичним.

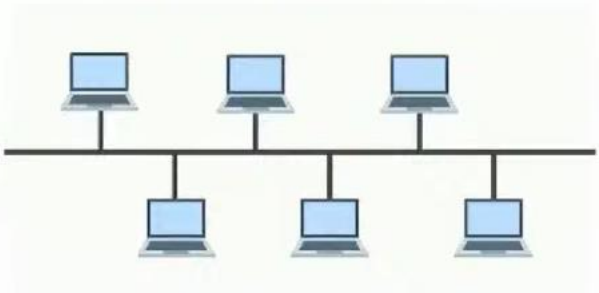

Рисунок 1.2 – Топологія шина

Топологія «кільце» − топологія, в якій кожен комп'ютер з'єднано лініями зв'язку лише з двома іншими (рис. 1.3): від одного він тільки отримує інформацію, а іншому тільки передає. Комп'ютери в «кільці» не є повністю рівноправними: одні обов'язково отримують інформацію від комп'ютера, який надсилає повідомлення в цей момент, раніше, а інші − пізніше.

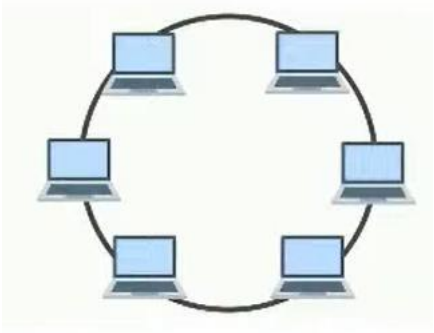

Рисунок 1.3 – Топологія кільце

У топології «зірка» всі комп'ютери мережі приєднано до центрального вузла (рис. 1.4), через який весь обмін інформацією йде від одного комп'ютера до іншого. Як центральний вузол можуть виступати концентратор чи комутатор − таку топологію називать пасивною «зіркою», або потужний комп'ютер, на який покладається дуже велике навантаження,− таку топологію називають активною «зіркою».

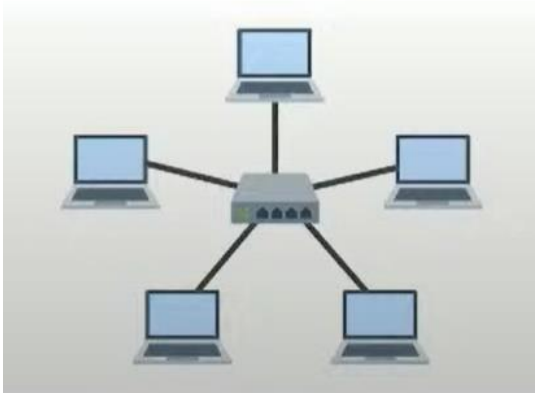

Рисунок 1.4 – Топологія зірка

Логічна топологія визначає як хости зв'язуються в мережі. Існує два основні види логічної топології: широкомовна (broadcast) та передача маркера (token passing) [2].

Broadcast в загальному розумінні означає, що кожен хост посилає дані, незалежно від інших хостів, в загальне мережеве середовище (моноканал), тобто випадково. Не існує порядку або правила, якому повинні слідувати робочі станції для використання мережі. Якщо першим відправив, то першим і обслуговується. Таким чином працює мережа Ethernet, яка розроблена компанією Xerox.

Другою логічною топологією є token passing. Token passing управляє доступом до мережі використовуючи електронні маркери, які передаються послідовно в одному напрямку від однієї робочої станції до іншої. Коли хост отримує маркер, то він починає передачу даних в мережу. Якщо, хост не готовий передавати інформацію, то у цьому випадку кадр маркера передається наступному хосту і процес знову повторюється. Прикладами мереж, які використовують маркерну передачу є : Token Ring (кільцева локальна мережа з маркерним доступом) та Fiber Distributed Data Interface (FDDI – оптоволоконний розподілений інформаційний інтерфейс, технологія побудови комп'ютерних мереж, що використовує для передачі сигналу оптоволоконний кабель).

Канали і лінії зв'язку. Кабельні системи. Характеристики ліній зв'язку.

Середовища передачі даних поділяються на [3]:

– середовища на мідній основі;

– оптоволоконні кабелі;

– безпровідні середовища.

До середовищ на мідній основі включають виту пару та коаксіальний кабель.

Вита пара − вид мережевого кабелю, з однією або декількома парами ізольованих провідників, скручених між собою (з невеликою кількістю витків на одиницю довжини) для зменшення взаємних наведень при передачі сигналу і покритих пластиковою оболонкою. Використовується для побудови мереж у багатьох технологіях, наприклад, Ethernet, ARCNet і Token ring. Останнім часом, завдяки своїй дешевизні й легкості установки, є найпоширенішим для побудови локальних мереж.

Підтримує передачу даних на відстань до 100 метрів. На більших відстанях сигнал через загасання не розпізнається; якщо передача даних на більшу відстань все ж таки необхідна, потрібно скористатися повторювачем, або ж задіяти коаксіальний кабель.

Залежно від наявності захисту − електрично заземленої мідної сітки або алюмінієвої фольги навколо скручених пар, визначають різновиди цієї технології:

1. Екранована вита пара (Shielded twisted pair, STP). Для захисту сигналу від шумів та спотворень використовується ефект взаємокомпенсації, екранування та попарне скручення проводів. Фізичні характеристики такого кабелю наступні:

– хвильовий опір − 150 Ом;

– пропускна здатність − до 100 Мб/с;

– рекомендована довжина фізичного сегменту − до 100 м.

Такий кабель забезпечує добрий захист від електромагнітних та радіочастотних наводок, але є порівняно дорогим та важким у прокладанні.

2. Екранована вита пара (Screened twisted pair, ScTP, Foiled twisted pair, FTP).

Володіє практично такими ж захисними властивостями, як і попередній. Фізичні характеристики:

– хвильовий опір 100-120 Ом;

– пропускна здатність до 100 Мб/с;

– рекомендована довжина фізичного сегменту − до 100 м.

Обидва вищеописані типи кабелю вимагають, щоб екрани були добре заземлені на обох кінцях, інакше замість екранування вони починають підсилювати зовнішні шуми.

3. Неекранована вита пара (Unshielded twisted pair, UTP).

Для зменшення впливу як зовнішніх так і внутрішніх шумів покладається лише на ефект взаємокомпенсації та попарне скручення проводів. Фізичні характеристики:

– хвильовий опір − 100 Ом;

– пропускна здатність − 100 і більше Мб/с (залежно від категорії кабелю);

– максимальна рекомендована довжина фізичного сегменту − до 100 м.

Перевагами використання цього кабелю є його дешевизна та легкість у прокладанні; недоліками − неможливість використання у зашумленому та агресивному середовищі.

Існує декілька категорій кабелю вита пара, які нумеруються від CAT1 до CAT7. Кабель вищої категорії зазвичай містить більше пар дротів і кожна пара має більше витків на одиницю довжини. Категорії неекранованої витої пари описуються в стандарті EIA/TIA 568 (Американський стандарт проводки в комерційних спорудах):

– CAT1 − телефонний кабель, всього одна пара. В США використовувався раніше, і провідники були скручені між собою. Використовується тільки для передачі голосу або даних за допомогою модему;

– CAT2 − старий тип кабелю з 2-х пар провідників, підтримував передачу даних на швидкостях до 4 Мбіт/с, використовувався в мережах token ring і ARCNet. Зараз іноді зустрічається в телефонних мережах;

– CAT3 − 2-парний кабель, використовувався для побудові локальних мереж 10BASE-T і token ring, підтримує швидкість передачі даних тільки до 10 Мбіт/с. На відміну від попередніх двох, відповідає вимогам стандарту IEEE 802.3. Також дотепер зустрічається в телефонних мережах;

– CAT4 − кабель складається з 4-х скручених пар, використовувався в мережах token ring, 10BASE-T, 10BASE-T4, швидкість передачі даних не перевищує 16 Мбіт/с, зараз не використовується;

– CAT5 − 4-парний кабель, це і є те, що зазвичай називають кабель «вита пара». Завдяки високій швидкості передачі (до 100 Мбіт/с при використанні 2 пар і до 1000 Мбіт/с при використанні 4 пар) є найпоширенішим мережевим носієм, що використовується в ком-п'ютерних мережах дотепер. Для прокладки нових мереж користуються дещо вдосконаленим кабелем CAT5e, який краще пропускає високочастотні сигнали;

– CAT6 − Застосовується в мережах Fast Ethernet і Gigabit Ethernet, складається з 4 пар провідників і здатний передавати дані на швидкості до 10000 Мбіт/с. Доданий до стандарту в червні 2002 року, пропускає сигнали частотою до 200МГц. Існує категорія CAT6e, в якій збільшена частота сигналу, що пропускається, до 500МГц. За даними IEEE, 70% встановлених мереж у 2004 році використовували кабель категорії CAT6, проте, можливо, це просто данина моді, бо й кабелі CAT5 і CAT5e цілком справляються в мережах 10GBASE-T;

– CAT7 − Специфікація на цей тип кабелю поки що не затверджена, швидкість передачі даних − до 10000 Мбіт/с, частота сигналу, що пропускається, до 600-700 Мгц. Кабель цієї категорії екранований.

Вита пара широко застосовується в мережевих технологіях і комунікаціях; кабелем категорії 6 замінюють коаксіальний кабель. Незважаючи на велику захищеність екранованої витої пари, вона не набула широкого поширення через складність в установці − необхідне заземлення і кабель, порівняно з неекранованою звитою парою, жорсткіший.

Коаксіальний кабель

Коаксіальний кабель − електричний кабель із співвісними провідниками. На даний момент цей кабель вже досить рідко застосовується для прокладання комп'ютерних мереж, хоча широко використовується для інших технологій передачі даних (телебачення) [3].

Фізичні характеристики:

– хвильовий опір − 50 Ом;

– пропускна здатність − до 100 Мб/с;

– рекомендована довжина сегменту − до 185 м (тонкий коаксіал) та до 500 м (товстий коаксіал).

Мідне оплетення кабелю одночасно виступає і захисним екраном для центрального провідника, і другим провідником у кабелі. Він є досить дешевим, але вже не задовольняє сучасних вимог до комп'ютерних мереж через недосконалість фізичної топології, яку можна на ньому реалізувати.

Оптоволоконний кабель [3] служить середовищем передачі даних для модульованого електромагнітного випромінювання із певною, строго визначеною довжиною хвилі (світлових імпульсів). Його основною перевагою є значна швидкість передачі даних (до 10 Гб/с) та довжина фізичного сегменту (до 40 км) порівняно із середовищами на мідній основі, а також несприйнятливість до зовнішніх електромагнітних шумів. Однак цей кабель є значно дорожчим порівняно з іншими, а також більш складним у прокладанні.

Принципи передачі сигналу в оптоволоконному середовищі. Світло, яке є носієм сигналу у оптоволоконному середовищі − це один з видів електромагнітної енергії. Як відомо, ця енергія у формі хвиль може проходити через вакуум, повітря та через деякі матеріали − наприклад, скло. Важливою характеристикою будь-якої енергетичної хвилі є довжина. Довжина електромагнітної хвилі визначається частотою коливання електричного заряду, який генерує цю хвилю.

Для передачі інформації через оптоволокно використовуються електромагнітні хвилі із довжинами, що лежать поза межами видимого діапазону (400-700 нм). Як правило, це хвилі довжиною 850 нм, 1310 нм або 1550 нм. Ці довжини були вибрані, оскільки хвилі з такими параметрами проходять через оптоволокно краще, ніж хвилі з іншими параметрами.

Виходячи з джерела, електромагнітні хвилі розповсюджуються по прямій. Ці прямі лінії називають променями. У вакуумі світлові промені розповсюджуються на швидкості 300000 км/с. Але у середовищі (вода, скло) ці швидкості є меншими.

Коли світловий промінь потрапляє на межу розділу двох середовищ (падаючий промінь), частина світлової енергії відбивається назад (відбитий промінь). Та частина світлової енергії, яка не відбилася, буде поглинута іншим середовищем. Але через різницю оптичної густини падаючий промінь заломиться. Саме завдяки заломленню світлових променів на межі розділу середовищ можливе використання оптоволоконного кабелю для передачі інформації. Кут заломлення світлового променя залежить від оптичної густини матеріалу. Оптична густина визначає, наскільки швидкість розповсюдження світла у середовищі менша від швидкості розповсюдження світла у середовищі. Відношення швидкості світла у середовищі до швидкості світла у вакуумі називається індексом заломлення. Отже, мірою оптичної густини матеріалу є його індекс заломлення. Збільшити індекс заломлення матеріалу (наприклад, скла) можна, додаючи до нього певні хімічні елементи. Якщо падаючий промінь падає на межу розділу двох середовищ під кутом 90, промінь не заломлюється. Але якщо кут відмінний від 90, промінь заломлюється, причому кут заломлення залежить як від індексу заломлення середовищ, так і від кута падіння променя. Якщо світловий промінь переходить із середовища з меншим індексом заломлення у сере-довище з меншим індексом заломлення, заломлений промінь загинається у сторону нормалі. Якщо ж навпаки − заломлений промінь загинається у протилежний до нормалі бік. Кут падіння, при якому промінь при переході з більш оптично густого середовища у менш оптично густе вже не заломлюється, а повністю відбивається у середовище, називається критичним кутом.

Світловий промінь, який несе інформацію у оптоволокні, мусить залишатися всередині оптоволокна на всьому шляху від відправника інформації до отримувача. Він не повинен заломлюватися всередину матеріалу, який знаходиться навколо світловоду, оскільки через заломлення буде втрачатися частина енергії.

Закони відбивання та заломлення ілюструють, як спроектувати волокно, у якому світлова енергія буде втрачатися мінімально. Таке волокно повинно задовольняти двом умовам:

– центральна частина оптоволокна повинна мати більший індекс заломлення, ніж матеріал, який її оточує;

– кут падіння світлового променя повинен бути більшим за крити-чний кут для ядра та оболонки.

– Коли обидві ці умови виконуються, падаючий промінь повністю за-лишається у волокні. Це явище називається повним внутрішнім відбиван-ням.

– Першу умову виконати легко, підібравши відповідним чином мате-ріали для ядра та оболонки. Контролювати кут падіння променя дозво-ляють два фактори:

– числова апертура − межі кутів падіння променя, при яких він буде повністю відбиватися;

– мода − шлях проходження променя через оптоволокно.

Отже, світлові промені можуть увійти в ядро лише у тому випадку, якщо кут падіння лежить у межах числової апертури волокна.

Будова і особливості застосування оптоволоконного кабелю

Якщо діаметр волокна дозволяє, можна одночасно пропустити через нього кілька променів. Говорять, що таке волокно є багатомодовим на відміну від одномодового, у якому може проходити лише один промінь у певний момент часу.

Кожен волоконно-оптичний кабель, який використовується для пе-редачі інформації у мережах, складається з двох світловодів у спільній оболонці − для передачі інформації у двох напрямках.

Оскільки у оптоволоконному кабелі не виникає проблем, пов’язаних з перехресними наводками, немає потреби екранувати або перекручувати пари проводів. Тому один кабель може нести від 2 до 48 світловодів.

Як правило, волоконно-оптичний кабель має наступну будову:

– зовнішня оболонка;

– підсилюючий матеріал;

– буфер;

– оболонка;

– ядро.

Ядро оптоволокна виготовляється із світло провідного матеріалу − як правило, це скло, виготовлене з двоокису кремнію та інших матеріалів. Багатомодове оптоволокно використовує тип скла, який називають ступінчасто індексованим склом . У такого скла індекс заломлення зменшується у напрямку до зовнішнього краю ядра.

Оболонка навколо ядра виготовляється також з двоокису кремнію, але з меншим індексом заломлення, ніж ядро. Це дозволяє досягнути у ядрі ефекту повного внутрішнього відбивання. Як правило, стандартне багатомодове оптоволокно використовує 50-ти або 62,5-мікронне ядро та 125-ти мікронну оболонку. Це позначається як 62,5/125 μ або 50/125 μ оптоволокно.

В якості буферизуючого матеріалу, як правило ,використовується пластик. Він дозволяє убезпечити оболонку та ядро від пошкоджень. Для цього існує 2 види дизайну кабелю: із вільним положенням ядра та із жорстко закріпленим ядром. Як правило, у LAN використовується багатомодовий кабель із жорстко закріпленим ядром. Він призначений для прокладки всередині будівель, тоді як кабель із вільним положенням ядра використовується для зовнішніх робіт.

Підсилюючий матеріал навколо буферизуючого шару попереджає ушкодження кабелю у процесі інсталяції. Для його виготовлення, як правило, використовують кевлар.

Зовнішня оболонка використовується для попередження забруднення кабелю розчинниками, абразивними речовинами та іншим.

У якості джерела випромінювання у багатомодовому оптоволокні використовуються інфрачервоні фотодіоди (Light Emitting Diodes, LEDs) або лазери.Багатомодове оптоволокно можна використовувати для пере-дачі інформації на відстань до 2 км.

У одномодовому оптоволокні (9/125 μ) у якості джерела використовується інфрачервоні лазери. Одномодове оптоволокно можна використовувати для передачі інформації на відстань до 40 км.

Оскільки інформація у мережних вузлах представлена у вигляді електричних сигналів, необхідна наявність пристою для перетворення електричних сигналів у оптичні і навпаки. Для цього використовуються:

– фотодіоди, які можуть генерувати хвилі з довжиною 850 нм або 1310 нм. Для фокусування світла у ядро використовуються лінзи;

– лазери, які генерують вузький промінь когерентного випромінювання з довжиною 1310 нм або 1550 нм.

Для прийому інформації і її зворотнього перетворення використовуються фотоприймачі. Вони приймають світлові імпульси з чітко визначеною довжиною хвилі та конвертують їх у електричні сигнали.

Для приєднання кабелю до портів мережних пристроїв використовуються конектори: з багатомодовим оптоволокном − Subscriber Connector (SC-коннектор), з одномодовим − Straight Tip (ST-коннектор) (рис. 1.5).

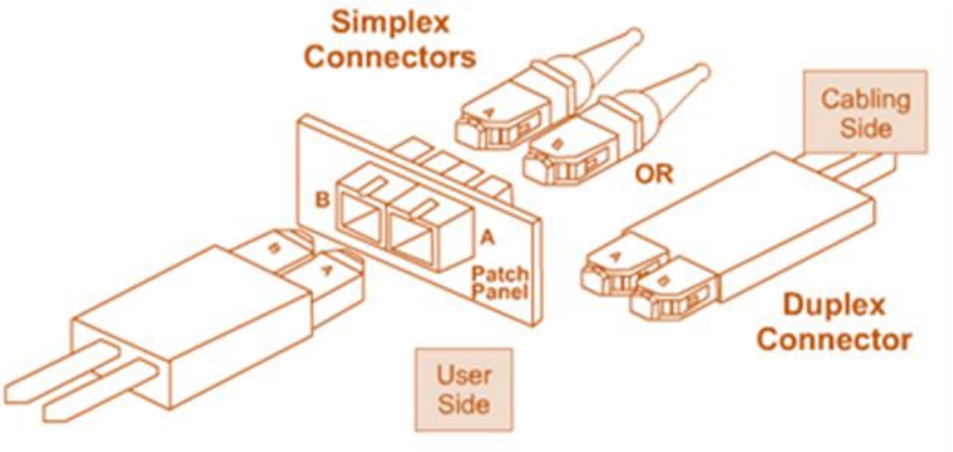

Рисунок 1.5 – Конектори

Мережеві пристрої

Мережеве обладнання − пристрої, необхідні для роботи комп'ютерної мережі. Наприклад: маршрутизатор, комутатор, концентратор, патч-панель та ін. Зазвичай розрізняють активне та пасивне мережеве обладнання.

Активне мережеве обладнання має певні «інтелектуальні» можливості. До цього типу належать маршрутизатор, комутатор (світч).

Під пасивним мережним устаткуванням мається на увазі обладнання, не наділене «інтелектуальними» особливостями. Таким обладнанням вважається кабельна система, вилка/розетка, повторювач, патч-панель, концентратор (хаб), монтажні шафи, стійки.

Мережеві пристрої забезпечують транспортування даних між пристроями користувача. Вони подовжують і об'єднують кабельні з'єднання, перетворюють дані з одного формату в інший і керують передаванням даних.

До мережевих пристроїв належать:

– повторювач (англ. repeater) − це пристрій, призначений для підсилення мережевих сигналів, що дозволяє передавати їх середовищем на більшу відстань. Причому повторювач не переглядає іншу інформацію, яка міститься в пакеті;

– концентратор (англ. hub − центр уваги) − це один із видів мережевих пристроїв, які можна встановлювати на рівні доступу мережі Ethernet. На ньому є кілька портів для під'єднання вузлів до мережі;

– концентратор не визначає, якому вузлу призначено конкретне повідомлення. Він просто приймає електронні сигнали одного порту й відтворює їх для всіх інших портів. Для передавання та отримання повідомлень всі порти концентратора Ethernet під'єднуються до одного і того самого каналу;

– міст (англ. bridge − міст) − це пристрій, призначений для фільтрування потоків даних у локальній мережі для того, щоб локалізувати передавання даних і разом із тим зберегти можливість зв'язку з іншими частинами мережі для перенаправлення туди потоків даних. Міст збирає інформацію про те, на якому порті знаходиться конкретна MAC-адреса, і приймає рішення про пересилку даних на підставі відповідного списку MAC-адрес. Мости здійснюють фільтрацію потоків даних, базуючись лише на MAC-адресі вузлів, тому можуть швидко пересилати дані;

– комутатор (англ. switch − перемикач) − це пристрій, який можна назвати «розумним» концентратором, тому що він передає дані тільки безпосередньо отримувачу;

– маршрутизатори (англ. router) − це пристрої об'єднаних мереж, які пересилають пакети між мережами на основі адрес. Маршрутизатор здатний вибирати найкращий шлях у мережі для переданих даних.

Маршрутизатор може приймати рішення на основі мережевих адрес замість використання індивідуальних MAC-адрес другого рівня. Завдяки цій здатності маршрутизатори стали основною магістраллю глобальної мережі Internet.

Мережева карта (мережевий інтерфейс) − пристрій, яким оснащують комп'ютер для під'єднання до мережі за допомогою мережевого кабелю чи радіоканалу. Для під'єднання до бездротової мережі можуть використовуватися не тільки мережеві карти, а й спеціальні пристрої.

Мережеві інтерфейси виготовляють у вигляді плат або окремих пристроїв − для бездротових мереж. Тип мережевого інтерфейсу має відповідати типу середовища передавання.

Тема 2. Адресація в мережах

Адресація в комп'ютерних мережах: поняття адресації у мережах, роль адрес у маршрутизації та ідентифікації пристроїв. Фізична (MAC) адресація: призначення MAC-адреси, структура MAC-адреси (OUI та індивідуальна частина), особливості роботи на канальному рівні. Логічна адресація (IP-адреси): IPv4: структура та поділ на мережеву і хостову частину, класи IP-адрес та історичний підхід. IPv6: структура, скорочений запис, переваги. Мережеві маски та підмережі: призначення маски, CIDR (Classless Inter-Domain Routing), приклади розбиття на підмережі. Спеціальні та зарезервовані адреси: приватні адреси (RFC 1918), петльова адреса (loopback), широкомовні та мультикаст-адреси. Протоколи підтримки адресації: ARP (Address Resolution Protocol) у IPv4, NDP (Neighbor Discovery Protocol) у IPv6, DHCP як механізм автоматичного призначення адрес. Імена і служби адресації в Інтернеті: DNS як система відображення імен у IP-адреси, ієрархічна структура доменів, організації, що керують адресним простором (IANA, ICANN, RIR).Проблеми та виклики адресації: вичерпання IPv4, перехід на IPv6, використання NAT і PAT як тимчасове рішення. Принципи роботи та застосування NAT та PAT. Причини появи NAT: обмеженість IPv4-адресного простору.

Адресація в комп'ютерних мережах

Спробуємо розібратися, які саме вимоги можуть висуватися до мережевих адрес. Наприклад, було відправлено поштовий переказ на значну суму. Напевне, менш за все би хотілося, щоб у місті існувала ще одна вулиця, будинок, квартира і отримувач з точно такими ж даними, як Ваші. Тобто головною і обов'язковою вимогою до будь-якої адреси є її унікальність. Інакше грошовий переказ або дані можуть бути доставлені зовсім не тому, кому призначались. Також необхідно враховувати, що адреса передається разом з даними, тобто чим більше місця займатиме адреса, тим менше місця залишиться для даних. Окрім того, короткі адреси простіше аналізувати комутаційному обладнанню, тому наступною вимогою до адрес можна вважати компактність [3-7].

Крім мережевого обладнання і обчислювальних пристроїв, адреси використовуються людьми. Поглянувши на дві адреси - ukr.net та 212.42.76.253, − зрозуміємо, що перша буде більш зручною для запам'ятовування. А отже, ще однією вимогою можна вважати зручність.

Уявімо, що адресою кожної окремої людини, наприклад в Україні, буде його ідентифікаційний код. Ідентифікаційний код є унікальним – розраховується за певним алгоритмом і не може повторюватися, компактним – складається з десяти цифр, частково зручним – так, можливо, запам'ятовувати коди всіх знайомих було б складно, проте цифровий код досить просто опрацьовувати. Проте як має виглядати доставка, наприклад, посилки, за такою адресою, враховуючи, що за ідентифікаційним кодом не можна вказати, в якому місті чи селищі мешкає людина? Якщо ж поглянути на адресу, яка використовуються для доставки пошти, то можна побачити, що вона складається з послідовних уточнень: країна, місто, вулиця, будинок. Тобто коли, наприклад, посилка відправляється з Франції, то перш за все вона направляється в країну доставки, потім відвозиться до певного міста. Якщо в місті декілька поштових відділень, то за назвою вулиці посилка направляється до відповідного поштового відділення і т.д. Можливість такого поетапного пошуку забезпечується існуванням ієрархічності адреси.

Отже, серед вимог, що висуваються до мережевих адрес, можна виділити:

– унікальність;

– компактність;

– зручність;

– ієрархічність.

Аналізуючи визначені вимоги, можна бачити, що деякі з них погано узгоджуються між собою. Так, наприклад, вимога унікальності і компактності можуть суперечити одна одній: максимально компактна адреса буде складатися з одного символу, але таких адрес буде досить мало. Така сама ситуація з компактністю та ієрархічністю – чим більш компактна адреса, тим менше нею може забезпечуватися ієрархічність.

Через подібні особливості в мережевих технологіях одночасно існують і використовуються різні адреси:

– фізичні, локальні, апаратні адреси (Physical, Local, Hardware Addresses);

– логічні, мережні адреси (Logical, Network Addresses);

– символьні, текстові адреси (Symbolic, Text Addresses).

Проте слід розуміти, що лише одна вимога має основне значення, і це − унікальність. Всі інші – не більш ніж побажання.

Для ідентифікації людини можна використати її відбитки пальців та поштову адресу. Зазвичай, відбитки пальців людини не змінюються, та за їх допомогою можна фізично ідентифікувати людину, де б вона не знаходилися. Інша справа − поштова адреса людини, яка залежить від місця її проживання, отже, може змінюватися упродовж життя.

Пристрої, що підключені до мережі, мають принаймні дві адреси, які аналогічні відбиткам пальців людини і її поштовій адресі. Це два типи адрес:

– MAC-адреса (Media Access Control) – адреса управління доступом до середовища передавання даних);

– IP-адреса (Internet Protocol) – адреса Інтернет-протоколу.

Мережевим вузлам потрібні обидві адреси для обміну даними мережею. MAC-адреса не змінюється при переміщенні пристрою з однієї мережі в іншу, оскільки вона призначається виробником мережевого інтерфейсу. IP-адреса може змінюватися в залежності від під'єднання пристрою до певної мережі, та призначається адміністратором мережі або відповідними службами мережі.

MAC-адреса (media access control address) – це унікальний ідентифікатор, що має мережевій адаптер, та застосовується у процесі передачі даних у межах локальної мережі (окремого канального сегменту мережі).

MAC-адреса має довжину 48 біт (6 байт). Для подання MAC-адреси використовується шістнадцятковий формат. Інших обмежень щодо подання не висувається, тому можна зустріти різні записи MAC-адрес, які відрізняються групуванням байтів та роздільними знаками:

– 00-50-56-BE-D7-87 – формат запису IEEE EUI-48;

– 00:50:56:BE:D7:87 – формат запису Unix Zero-Padded. 0050.56BE.D787 – формат запису Cisco.

Історично адреси прошивалися в ПЗУ чіпсету мережевої карти без можливості їх модифікації, але нині MAC-адреса може бути змінена програмно.

MAC-адреса складається з двох частин. Перша частина MAC-адреси вказує постачальника-виробника мережевого інтерфейсу. Ця частина MAC-адреси називається унікальним ідентифікатором організації (OUI – Organizationally Unique Identifier). Довжина OUI найчастіше складає 3 байти (24 біти), але може бути і 28 або 36 біт. Керування загальним адресним простором MAC-адрес здійснює Інститут інженерів електриків та електронників (IEEE – Institute of Electrical and Electronics Engineers). Отже, постачальник, який бажає виготовляти і продавати мережеві інтерфейси, повинен зареєструватися в IEEE, щоб йому надали ідентифікатор OUI.

Друга частина адреси (біти, що залишилися) − це унікальний ідентифікатор інтерфейсу (OUA – Organizationally Unique Address). Всі MAC-адреси, що починаються з однакового ідентифікатора OUI, повинні містити унікальні ідентифікатори інтерфейсів.

Тому в теорії MAC-адреси унікальні (подвійно унікальні), оскільки кожен з виробників зобов'язаний забезпечувати унікальність адреси для кожного виробленого ним пристрою. Однак деякі виробники для OUA встановлюють випадкове число, що може призводити до їх дублювання.

IP-адреса (Internet Protocol address) – це ідентифікатор, що призначається мережному адаптеру/інтерфейсу і використовується для адресації комп'ютерів чи пристроїв у мережах, побудованих з використанням протоколу TCP/IP. Важливою особливістю IP-адрес є їх ієрархічність, тобто IP-адреса ґрунтуючись на розміщення пристрою в мережі.

Існують четверта та шоста версії IP-адресації. Основним стандартом, у якому описуються вимоги до IP-адрес версії 4, є прийнятий у вересні 1981 року стандарт RFC-791

«Internet Protocol. DARPA Internet Program Protocol Specification». Основним стандартом, у якому описуються вимоги до IP- адрес версії 6, є прийнятий у грудні 1998 року стандарт RFC-2460 «Internet Protocol, Version 6 (IPv6) Specification». Пізніше ці стандарти були доповнені іншими стандартами RFC, що певною мірою стосуються питань IP-адресації. Тексти стандартів RFC, зокрема і зазначених вище стандартів, можна отримати на Web-сайті організації, що займається стандартизацією – Підрозділу інженерних розробок Інтернет (IETF, Internet Engineering Task Force).

IPv4: 32-бітна адреса. Записується в десятковому форматі чотирма числами, розділеними крапками. Наприклад, 192.168.10.10 (рис. 2.1).

IPv6: 128-бітна адреса. Записується в шістнадцятковому форматі (рис. 2.2).

Наприклад, 2001:0DB8:0000:ABCD:0000:0000:0000:1234.

Незважаючи на те, що IPv4-адреса записується в десятковому форматі, її опрацювання здійснюється в двійковому. Кожне число, відокремлене крапкою, називається октетом («ОКТО» − вісім), тому що містить 8 біт.

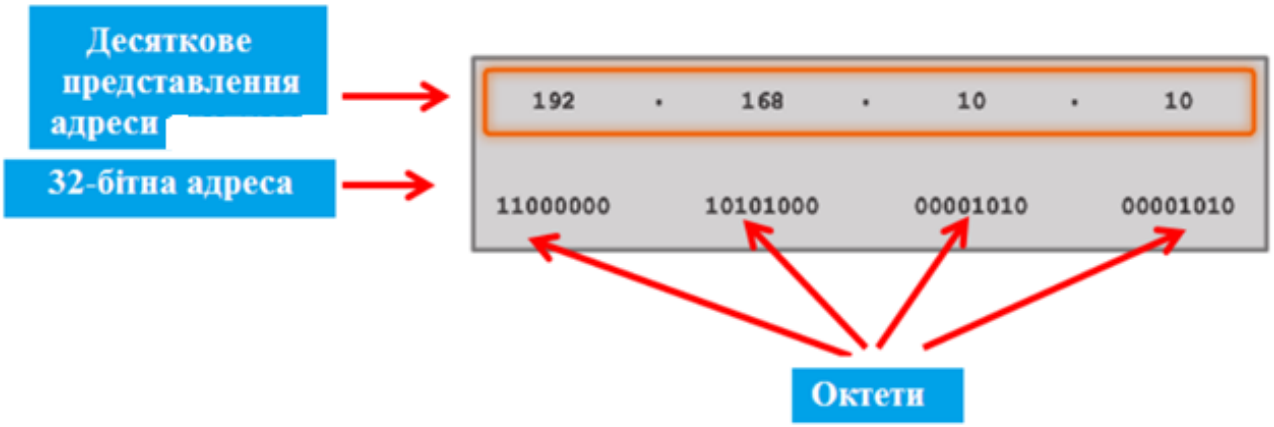

Рисунок 2.1 – Формат запису IPv4-адреси [5]

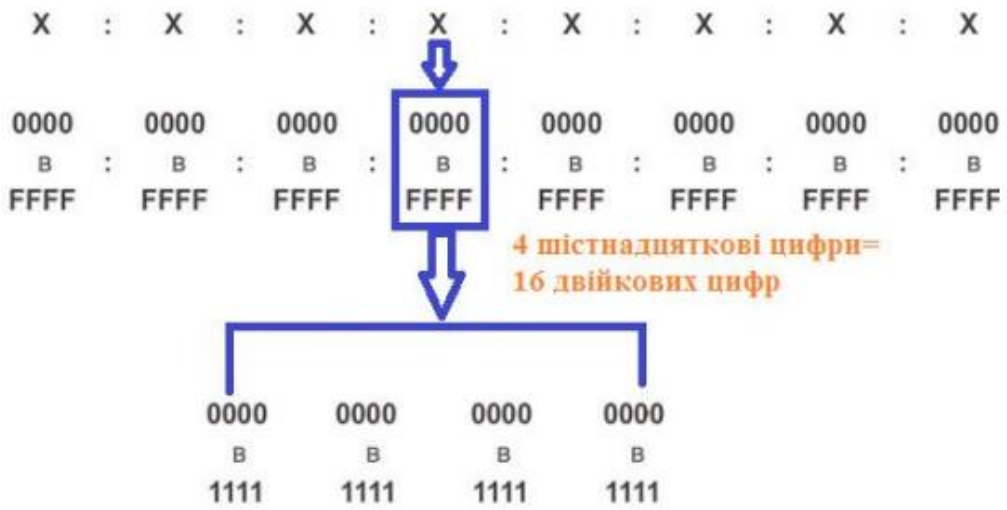

Рисунок 2.2 – Формат запису IPv6-адреси [5]

Таким чином, адреса 192.168.10.10 складається з чотирьох октетів. Кожен біт в октеті може бути 1 або 0. Тому кожен октет (8 біт) може містити десяткове значення від 0 до 255 включно (від 00000000 до 11111111).

Призначення мережевої маски

Мережева маска визначає, яка частина IP-адреси належить до мережі, а яка − до вузла (host).

Вона використовується для:

– ідентифікації межі між мережевим та хостовим простором;

– забезпечення маршрутизації пакетів у глобальній мережі;

– оптимізації використання адресного простору.

Наприклад, маска 255.255.255.0 (або /24) означає, що перші 24 біти адреси визначають мережу, а останні 8 − хост.

CIDR (Classless Inter-Domain Routing)

CIDR було запроваджено для подолання обмежень класової адресації (A, B, C).

Основні переваги CIDR:

– гнучкість у визначенні розміру підмережі;

– зменшення таблиць маршрутизації завдяки агрегації маршрутів;

– раціональне використання IPv4-адресного простору, що стало критично важливим у період його виснаження.

CIDR записується у форматі IP/префікс, де префікс вказує кількість бітів мережевої частини. Наприклад: 192.168.1.0/26.

Приклади розбиття на підмережі

Приклад 1: розбиття мережі /24. Мережа: 192.168.1.0/24 (256 адрес). Якщо розділити на підмережі /26: кожна підмережа матиме 64 адреси (62 доступні для хостів). Підмережі: 192.168.1.0/26, 192.168.1.64/26, 192.168.1.128/26, 192.168.1.192/26.

Приклад 2: Використання VLSM (Variable Length Subnet Masking). VLSM дозволяє застосовувати різні маски в межах однієї мережі. Наприклад, у мережі 10.0.0.0/24 можна виділити: /28 для невеликих сегментів (16 адрес), /30 для точка-точка з'єднань (4 адреси), /26 для більших сегментів (64 адреси).

Еволюція та сучасні тенденції:

– перехід від класової адресації до CIDR був ключовим етапом розвитку Інтернету;

– CIDR та VLSM забезпечили масштабованість та ефективність маршрутизації, особливо в умовах дефіциту IPv4;

– у сучасних мережах CIDR використовується також для IPv6, де префікси дозволяють створювати ієрархічну структуру адресного простору.

Мережеві маски є базовим інструментом для визначення структури IP-мережі:

– CIDR забезпечує гнучке та ефективне управління адресами;

– поділ на підмережі дозволяє оптимізувати використання ресурсів та підвищити безпеку мережі;

– використання VLSM дає можливість адаптувати розмір підмережі до конкретних потреб.

Приватні та публічні адреси

IPv4 використовує 32-бітну адресацію, що теоретично дозволяє створити приблизно 4,3 мільярда унікальних IP-адрес. На початковому етапі розвитку Інтернету така кількість вважалася надлишковою, однак історичні особливості розподілу адрес, зокрема класова адресація та виділення великих блоків організаціям без оптимального використання, суттєво зменшили ефективність цього простору. Згодом перехід до безкласової адресації CIDR лише частково розв'язав проблему, але не усунув її першопричину − обмежену довжину адреси [3].

IPv4 адреси бувають двох типів: публічні (public) і приватні (private) (рис. 2.3). Перші так само часто називають білими або зовнішніми, а другі − сірими або внутрішніми. У чому між ними різниця?

Публічні адреси є унікальними і не можуть повторюватися ніде і ніколи, що контролюється провайдером, який вам їх дав в оренду, а йому, у свою чергу дав їх в оренду інший провайдер або організація IANA, що стежить за розподілом адрес. А приватні, навпаки, можуть використовуватися ким завгодно і повторюватися скільки завгодно разів. Тобто, приватні адреси можуть повторюватися і не бути унікальними, але якою ціною? На них нічого не можна відправити з інтернету. Ніхто в інтернеті не знає маршруту до ваших приватних адрес, а так само що можливо тим же самими адресами, що використовується у вашого сусіда, а раз ніхто не знає, то яка різниця, повторюються вони чи ні? Іншими словами, за межами локальної мережі приватні адреси не маршрутизуются [4].

Характеристика	Приватні IP	Публічні IP
Унікальність	В межах мережі	Глобально
Доступність	Локальна мережа	Інтернет
Вартість	Безкоштовні	Платні

Рисунок 2.3 – Порівняння приват них і публічних адрес [5]

Приватних адрес не так багато − навіщо робити багато, якщо їх можна повторювати скільки завгодно разів (у різних локальних мережах, звичайно). Всього три діапазони приватних адрес (рис. 2.4) [5].

Клас A	10.0.0.0 - 10.255.255.255
Клас B	172.16.0.0 - 172.31.255.255
Клас C	192.168.0.0 - 192.168.255.255

Рисунок 2.4 – Діапазон приватних адрес [5]

Природно, що навряд чи знадобляться в чистому вигляді мережі таких великих розмірів, тому приватні адреси зазвичай розбивають на підмережі за допомогою більш довгого префікса, наприклад, з третього діапазону можна отримати 255 приватних підмереж по 254 хоста у кожній (рис. 2.5-2.6).

Наступне важливе питання − припустимо, ми видали співробітникам в офісі багато приватних адрес, але як же вони зможуть вийти в інтернет? Вони будуть відправляти запити в мережу, а відповіді на ці запити повинні будуть повертатися на зворотні адреси, які в даному випадку приватні. Так як в інтернеті ніхто не знає маршрутів до приватних адрес, то це не можливо. Як правило така задача вирішується одним із двох способів:

1) у мережі використовується проксі-сервер. Цей сервер має інтерфейс у зовнішній мережі і може мати такий же інтерфейс в приватній. Користувачі звертаються до нього, а не до сайтів безпосередньо. Сервер «свій», тому він знає про свої приватні адреси. Він отримує з них запити, і для кожного запиту звертається в інтернет зі своєї публічної адреси. Коли він отримає відповідь, то перешле її в середину на приватну адресу запитувача;

2) на граничному маршрутизаторі можна налаштувати трансляцію адрес (NAT). І тоді при проходженні пакета з локальної мережі в інтернет, адреса відправника буде змінюватися: замість нікому невідомої приватної вписуватиметься публічна адреса з деякого пулу адрес, або публічна адреса самого маршрутизатора (тут можливі різні реалізації). На цю адресу і будуть приходити відповіді з інтернету. У відповідях відбуватиметься зворотня заміна: публічна адреса одержувача буде замінюватися на вихідну приватну адресу, після чого пакет повертається клієнтові, який робив запит.

Двійкове значення октету	Значення бітів октету	Десяткове значення октету
00000000	0	0
10000000	128	128
11000000	128+64	192
11100000	128+64+32	224
11110000	128+64+32+16	240
11111000	128+64+32+16+8	248
11111100	128+64+32+16+8+4	252
11111110	128+64+32+16+8+4+2	254
11111111	128+64+32+16+8+4+2+1	255

Рисунок 2.5 – Двійкова та десяткові значення деяких октетів

Двійкове значення октету	Значення бітів октету	Десяткове значення октету
11000101	128+64+0+0+0+4+0+1	197
11000110	128+64+0+0+0+4+2+0	198
11000111	128+64+0+0+0+4+2+1	199
11001000	128+64+0+0+8+0+0+0	200

Рисунок 2.6 – Приклад відповідності двійкового та десяткового значення октетів

На рисунку 2.7 показані зліва направо класи адрес − значення старших біт та десяткових значень першого октету в даному класі та доступна кількість мереж та вузлів, підтримуваних у даному класі.

Клас адреси	Старші біти	Діапазон десяткових значень першого октету	Доступна кількість мереж	Доступна кількість вузлів
Клас A	0	1–126	126	16 777 214
Клас B	10	128–191	16 384	65 534
Клас C	110	192–223	2 097 152	254

Рисунок 2.7 – Класи адрес та відповідні їм ідентифікатори мереж та вузлів

В адресах класу А перший октет представляє ідентифікатор мережі, в адресах класу В перші два актети використовуються для ідентифікатора мережі і нарешті в адресах класу С перші три октети використовується для ідентифікатора мережі. Таким чином кожну адресу можна розділити на два компоненти, як показано на рисунку 2.8-2.9.

Клас адреси	IP-адреса	Ідентифікатор мережі	Ідентифікатор вузла
Клас A	w.x.y.z	w	x.y.z
Клас B	w.x.y.z	w.x	y.z
Клас C	w.x.y.z	w.x.y	z

Рисунок 2.8 – Розподіл IP-адреси на компоненти відповідно до її класу

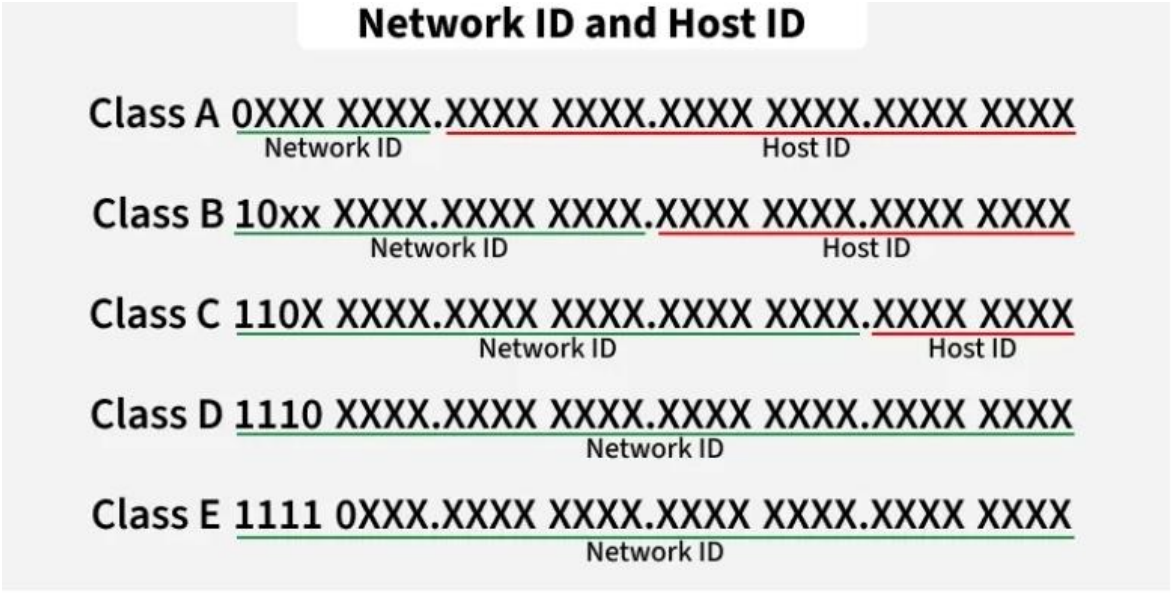

Рисунок 2.9 – Розподіл IP-адреси на компоненти відповідно до її класу [6]

Локальні адреси IPv4 і IPv6

Локальні адреси каналу для IPv4 і IPv6 використовуються пристроєм для зв'язку з іншими комп'ютерами, підключеними до однієї мережі в межах одного діапазону IP адрес. Основна відмінність IPv4 від IPv6 полягає в наступному:

– пристрій IPv4 використовує локальну адресу каналу, якщо не може отримати IPv4 адресу;

– у IPv6-пристрою завжди має бути динамічно або вручну налаштована локальна адреса каналу.

Якщо комп'ютеру з ОС Windows не вдається зв'язатися з DHCP сервером і отримати адресу IPv4, то ОС Windows автоматично призначає адресу засобами автоматичного призначення приватних IP-адрес (Automatic Private IP Addressing, APIPA). Локальні адреси каналу знаходяться в діапазоні від 169.254.0.0 до 169.254.255.255.

IPv6 локальна адреса каналу

Як і IPv4, локальна адреса каналу IPv6 дозволяє пристрою обмінюватися даними з іншими пристроями з підтримкою IPv6 в одній мережі і тільки в ній. На відміну від IPv4, кожен пристрій з підтримкою IPv6 повинен мати локальну адресу каналу. IPv6 локальні адреси каналу знаходяться в діапазоні від fe80:: до febf::.

Примітка: На відміну від локальних IPv4-адрес каналу, локальні IPv6-адреси каналу використовуються в різних процесах, включаючи протоколи виявлення мереж і протоколи маршрутизації.

Загальне поняття спеціальних і зарезервованих IP-адрес

У процесі функціонування комп'ютерних мереж поряд із звичайними унікальними IP-адресами, що призначаються мережевим інтерфейсам вузлів, використовуються спеціальні та зарезервовані адреси. Вони мають визначене призначення та застосовуються для тестування, керування, маршрутизації трафіку, організації групової передачі даних і забезпечення службових функцій мережевих протоколів.

Спеціальні IP-адреси не можуть бути довільно призначені кінцевим вузлам для звичайного обміну даними в мережі Інтернет. Їхня поведінка суворо регламентована стандартами IETF (RFC) і реалізується на рівні мережевих протоколів та операційних систем.

До найважливіших категорій спеціальних і зарезервованих адрес належать петльова (loopback) адреса, широкомовні (broadcast) адреси та мультикаст-адреси.

Петльова адреса (Loopback Address)

Петльова адреса використовується для внутрішньої перевірки роботи мережевого стеку на одному й тому самому вузлі без передавання даних у фізичну мережу. Фактично пакети, надіслані на loopback-адресу, повертаються назад у межах операційної системи [3-4].

У протоколі IPv4 для петльового інтерфейсу зарезервовано весь діапазон адрес

127.0.0.0/8, тобто від 127.0.0.0 до 127.255.255.255. Найчастіше на практиці використовується адреса 127.0.0.1, яка має стандартну назву localhost.

У протоколі IPv6 для аналогічних цілей використовується адреса ::1/128, яка також ідентифікується ім’ям localhost.

Петльова адреса виконує важливі функції:

– тестування мережевого програмного забезпечення;

– перевірка роботи TCP/IP-стеку без доступу до зовнішньої мережі;

– забезпечення взаємодії між локальними сервісами (наприклад, веб-сервером і базою даних на одному сервері);

– підвищення безпеки, оскільки трафік не виходить за межі хоста.

Важливою особливістю loopback-адрес є те, що пакети з такими адресами ніколи не маршрутизуються та автоматично відкидаються маршрутизаторами.

Широкомовні адреси (Broadcast Addresses)

Широкомовні адреси призначені для одночасного передавання мережевих пакетів усім вузлам у межах однієї логічної мережі або підмережі. Такий механізм застосовується для виявлення пристроїв, службової взаємодії та початкової конфігурації мережі.

У протоколі IPv4 існують два основні типи широкомовних адрес. Обмежена широкомовна адреса має вигляд 255.255.255.255. Вона використовується для передавання повідомлень усім вузлам локального сегмента мережі, коли відправник ще не знає параметрів мережі. Такі пакети не передаються через маршрутизатори. Спрямована широкомовна адреса формується як остання адреса в конкретній підмережі, тобто адреса, у якій усі біти хостової частини дорівнюють одиниці. Наприклад, для мережі 192.168.1.0/24 широкомовною буде адреса 192.168.1.255.

Широкомовна передача широко використовується в таких протоколах і службах:

– ARP (визначення MAC-адреси за IP-адресою);

– DHCP (отримання IP-адреси клієнтом);

– деякі протоколи маршрутизації та виявлення сервісів.

– У протоколі IPv6 класичне широкомовлення відсутнє. Його функціональність замінена мультикаст-механізмом, що дозволяє зменшити навантаження на мережу.

Мультикаст-адреси (Multicast Addresses)

Мультикаст-адреси використовуються для передавання даних не всім вузлам мережі, а лише певній групі отримувачів, які підписалися на відповідну мультикаст-групу. Такий підхід є значно ефективнішим за широкомовлення, особливо в мережах із великою кількістю пристроїв.

У протоколі IPv4 мультикаст-адреси належать до діапазону 224.0.0.0 – 239.255.255.255, що відповідає класу D у класовій адресації. Цей діапазон поділяється на піддіапазони з різним призначенням:

– адреси 224.0.0.0/24 використовуються для локальних мережевих протоколів і не маршрутизуються;

– адреси 239.0.0.0/8 призначені для приватного мультикасту в організаціях.

У протоколі IPv6 всі мультикаст-адреси починаються з префікса FF00::/8. В IPv6 мультикаст є основним механізмом групової доставки трафіку та використовується, зокрема, для заміни ARP, служби виявлення сусідів і роботи мережевих сервісів.

Мультикаст-передача застосовується в таких сценаріях:

– потокове відео та аудіо (IPTV, відеоконференції);
– протоколи маршрутизації (OSPF, RIP);
– служби синхронізації та розповсюдження оновлень;
– корпоративні системи сповіщення.

Значення спеціальних адрес для адміністрування та безпеки мереж

Неправильне використання широкомовлення або мультикасту може призвести до перевантаження мережі, а некоректна фільтрація таких адрес − до уразливостей безпеки.

Зокрема, в системах мережевої безпеки (брандмауери, IDS/IPS) широко застосовуються правила контролю broadcast- та multicast-трафіку, а loopback-інтерфейс часто використовується для ізольованого розміщення критично важливих сервісів.

ARP (Address Resolution Protocol) у мережах IPv4 та NDP (Neighbor Discovery Protocol) у мережах IPv6

Загальні засади адресації та необхідність протоколів виявлення сусідів

Функціонування комп'ютерних мереж на основі стеку TCP/IP базується на використанні логічної IP-адресації та фізичної адресації канального рівня, зокрема MAC-адрес. Передавання даних у локальній мережі Ethernet фізично здійснюється за MAC-адресами, тоді як прикладні та транспортні протоколи оперують IP-адресами. Це зумовлює необхідність механізму динамічного зіставлення IP-адреси вузла з його MAC-адресою. У мережах IPv4 таку функцію виконує протокол ARP, а в мережах IPv6 − протокол NDP, який є складовою частиною ICMPv6.

Протокол ARP у IPv4

ARP (Address Resolution Protocol) − це допоміжний протокол мережевого рівня, призначений для визначення MAC-адреси вузла за відомою IPv4-адресою в межах однієї широкомовної доменної мережі [8].

Коли вузол IPv4 потребує передати пакет іншому вузлу в тій самій локальній мережі, він спочатку перевіряє власну ARP-таблицю (ARP cache). Якщо відповідність між IP- та MAC-адресою відсутня, ініціюється процедура ARP-запиту. ARP-запит передається у вигляді широкомовного Ethernet-кадру, адресованого на MAC-адресу FF:FF:FF:FF:FF:FF, і містить IP-адресу шуканого вузла. Вузол, IP-адреса якого збігається з адресою в запиті, формує ARP-відповідь, що надсилається уніфіковано безпосередньо ініціатору запиту. Отримавши відповідь, вузол зберігає пару IP-MAC у ARP-таблиці на обмежений час, після чого може здійснювати безпосереднє передавання кадрів Ethernet. ARP працює без механізмів автентифікації та цілісності, що робить його вразливим до атак типу ARP spoofing або ARP poisoning. Такі атаки широко використовуються в локальних мережах для реалізації перехоплення трафіку, атаки «людина посередині» (Man-in-the-Middle) та відмови в обслуговуванні. Саме тому ARP має важливе значення в курсах з мережевої безпеки та реагування на інциденти.

Попри простоту реалізації, ARP має низку архітектурних недоліків. Основними з них є використання широкомовних повідомлень, що негативно впливає на масштабованість мережі, а також відсутність вбудованих механізмів безпеки. У процесі проєктування IPv6 ці обмеження були враховані, що зумовило відмову від ARP на користь більш функціонального та гнучкого механізму − Neighbor Discovery Protocol (NDP).

Neighbor Discovery Protocol (NDP) у IPv6

Neighbor Discovery Protocol − це набір процедур і повідомлень, реалізованих у межах ICMPv6, які забезпечують виявлення сусідніх вузлів, визначення їх канальних адрес, контроль досяжності, автоматичну конфігурацію адрес і виявлення маршрутизаторів.

На відміну від ARP, NDP не використовує широкомовлення. Усі повідомлення передаються за допомогою багатоадресної (multicast) адресації, що суттєво зменшує навантаження на мережу. Для зіставлення IPv6-адреси з MAC-адресою використовуються повідомлення Neighbor Solicitation та Neighbor Advertisement. За своєю логікою вони є аналогами ARP-запиту та ARP-відповіді, проте функціонують у рамках ICMPv6.

Окрім вирішення адрес, NDP виконує низку додаткових функцій. Протокол забезпечує виявлення маршрутизаторів за допомогою Router Solicitation і Router Advertisement, підтримує Stateless Address Autoconfiguration (SLAAC), а також реалізує механізм Neighbor

Unreachability Detection, який дозволяє визначати, чи доступний сусідній вузол у поточний момент часу. Таким чином, NDP поєднує функціональність ARP, ICMP Redirect та частково DHCP, що робить його ключовим елементом архітектури IPv6.

Хоча NDP є архітектурно досконалішим за ARP, він також не позбавлений вразливостей. Атаки типу Neighbor Spoofing, Rogue Router Advertisement або DoS через надмірні ICMPv6-повідомлення залишаються актуальними. Для підвищення рівня безпеки в IPv6 передбачено використання механізму Secure Neighbor Discovery (SEND), який базується на криптографічних підписах і сертифікатах, однак через складність реалізації він застосовується обмежено.

У практиці мережевого адміністрування безпека NDP часто забезпечується за допомогою фільтрації ICMPv6, контролю RA-повідомлень на комутаторах (RA Guard) та систем моніторингу мережевого трафіку.

Порівняльний аналіз ARP і NDP. ARP є простим, але обмеженим протоколом, призначеним виключно для зіставлення IPv4-адрес і MAC-адрес у локальній мережі. Він активно використовує широкомовлення та не має вбудованих засобів безпеки. NDP, навпаки, є комплексним механізмом IPv6, який використовує multicast, інтегрований з ICMPv6 та підтримує додаткові функції автоматичної конфігурації й виявлення маршрутизаторів.

З точки зору сучасних мереж, NDP краще відповідає вимогам масштабованості, автоматизації та керованості, однак потребує глибшого розуміння та ретельного налаштування з боку адміністратора.

ARP і NDP відіграють ключову роль у забезпеченні взаємодії між мережевим та канальним рівнями моделі TCP/IP. ARP залишається невід'ємною складовою IPv4-мереж, тоді як NDP є фундаментом функціонування IPv6.

Статична адресація

У невеликій мережі можна вручну налаштувати кожен пристрій з власною IP адресою. Ви призначаєте унікальну IP адресу кожному вузлу в одній мережі. Це процес відомий як статична IP адресація (рис. 2.10-2.11).

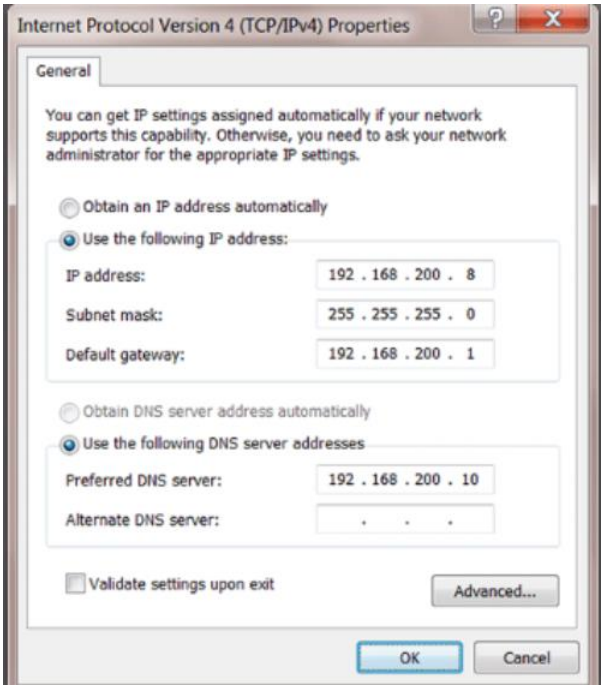

Рисунок 2.10 – Налаштування IPv4-адресації статично

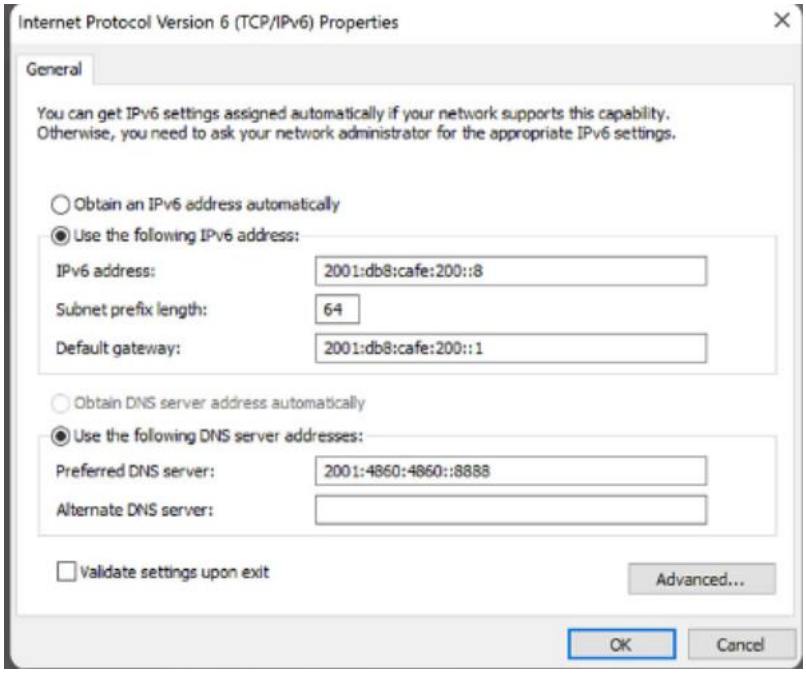

Рисунок 2.11 – Налаштування IPv6-адресації статично

На рисунку 2.11 показано, що на комп'ютері з ОС Windows можна призначити такі параметри IPv4 адреси вузла:

– IP адреса − ідентифікує пристрій у мережі;

– маска підмережі − використовується для ідентифікації мережі, до якої підключений даний пристрій;

– основний шлюз − визначає маршрутизатор, який цей пристрій використовує для доступу до Інтернету або іншої мережі

– необов'язкові параметри − наприклад, адреси основного та альтернативного DNS серверів

Динамічна адресація

Замість налаштування вручну кожного пристрою в мережі, можна скористатися перевагами реалізації сервера DHCP (Dynamic Host Configuration Protocol). DHCP сервер автоматично присвоює IP адреси, що спрощує процес адресації. Автоматичне налаштування деяких параметрів IP адресації також знижує можливість присвоєння повторних або недійсних IP адрес [9].

За замовчуванням більшість кінцевих пристроїв налаштовано на запит IP адреси з DHCP сервера. Налаштування за замовчуванням для комп'ютера з ОС Windows показані на рисунку. Якщо комп'ютер налаштовано на отримання IP адреси автоматично, усі інші поля IP адресації недоступні. Цей процес однаковий як для дротового, так і для бездротового мережевого адаптера.

DHCP сервер може автоматично встановлювати значення наступних налаштувань IPv4 адреси пристрою:

– IPv4 адреса;

– маска підмережі;

– основний шлюз (Default Gateway);

– необов'язкові параметри, наприклад адреса DNS сервера.

DHCP також доступний для автоматичного налаштування IPv6 адресації.

DHCP як механізм автоматичного призначення IP-адрес

У сучасних комп'ютерних мережах ефективне управління IP-адресним простором є критично важливим завданням, особливо в умовах великої кількості користувачів, динамічного підключення пристроїв та використання мобільних і віртуалізованих середовищ. Ручне призначення IP-адрес у таких умовах є неефективним, схильним до помилок і складним в адмініструванні. Саме тому широкого поширення набув протокол DHCP

(Dynamic Host Configuration Protocol), який забезпечує автоматичне, централізоване та контрольоване призначення параметрів мережевої конфігурації клієнтам.

DHCP − це мережевий протокол прикладного рівня моделі TCP/IP, призначений для автоматичного надання клієнтським пристроям IP-адреси та супутніх параметрів конфігурації. До таких параметрів належать маска підмережі, адреса шлюзу за замовчуванням, адреси DNS-серверів, доменне ім'я, час оренди адреси та інші опції. DHCP є розвитком більш раннього протоколу BOOTP і зберігає з ним зворотну сумісність.

Основною ідеєю DHCP є використання централізованого сервера або групи серверів, які керують пулом IP-адрес та розподіляють їх клієнтам на певний проміжок часу, що називається орендою (lease).

У моделі OSI DHCP функціонує на прикладному рівні, однак для своєї роботи використовує транспортний протокол UDP. Обмін повідомленнями відбувається через порт 67 UDP на стороні сервера та порт 68 UDP на стороні клієнта. Використання UDP обумовлене необхідністю швидкого та простого обміну без встановлення з'єднання, а також можливістю широкомовної передачі.

Робота DHCP базується на клієнт-серверній архітектурі. Коли пристрій підключається до мережі та не має IP-адреси, він ініціює процес отримання конфігурації. Класичний алгоритм взаємодії між клієнтом і сервером описується моделлю DORA, яка включає чотири основні етапи: Discover, Offer, Request, Acknowledgment. На першому етапі клієнт надсилає широкомовне повідомлення DHCP Discover з метою виявлення доступних DHCP-серверів у мережі. Це повідомлення передається без IP-адреси відправника, оскільки клієнт ще не має власної адреси. На другому етапі один або кілька DHCP-серверів відповідають повідомленням DHCP Offer, у яких пропонують клієнту доступну IP-адресу та інші параметри конфігурації. На третьому етапі клієнт обирає одну з отриманих пропозицій і надсилає повідомлення DHCP Request, підтверджуючи бажання отримати конкретну адресу від конкретного сервера. На завершальному етапі сервер надсилає повідомлення DHCP Acknowledgment (ACK), яким остаточно закріплює IP-адресу за клієнтом на визначений час оренди. Після цього клієнт може повноцінно функціонувати в мережі.

Оренда IP-адреси та її поновлення

IP-адреси в DHCP не призначаються назавжди, а надаються в оренду на певний час. Такий підхід дозволяє ефективно використовувати обмежений адресний простір. Протягом терміну оренди клієнт зобов'язаний періодично оновлювати її. Приблизно після половини часу оренди клієнт намагається поновити адресу, надсилаючи DHCP Request безпосередньо серверу, який її надав. Якщо сервер підтверджує поновлення, оренда продовжується без переривання мережевої роботи. У разі недоступності сервера клієнт може продовжити спроби поновлення або, після завершення терміну оренди, припинити використання IP-адреси та ініціювати новий цикл отримання конфігурації.

Статичні та динамічні призначення

DHCP підтримує як динамічне, так і умовно статичне призначення IP-адрес. Динамічне призначення передбачає видачу будь-якої вільної адреси з пулу. Умовно статичне призначення, відоме як резервування, базується на MAC-адресі клієнта: конкретному пристрою завжди видається одна й та сама IP-адреса. Такий підхід широко використовується для серверів, мережевих принтерів, IP-телефонів та іншого обладнання, яке повинно мати стабільну адресу, але при цьому керуватися централізовано.

DHCP Relay

У великих мережах, розділених на кілька підмереж або VLAN, широкомовні DHCP-повідомлення не проходять через маршрутизатори. Для вирішення цієї проблеми використовується механізм DHCP Relay. DHCP Relay-агент приймає широкомовні запити від клієнтів у локальній мережі та пересилає їх у вигляді унікаст-повідомлень на віддалений DHCP-сервер. Це дозволяє використовувати один або кілька централізованих серверів DHCP для обслуговування великої кількості сегментів мережі.

Безпека DHCP

Незважаючи на простоту та зручність, DHCP має низку вразливостей. Однією з основних загроз є атака типу rogue DHCP, коли в мережі з'являється несанкціонований

DHCP-сервер, що роздає некоректні або шкідливі параметри конфігурації. Іншою загрозою є DHCP starvation − атака, спрямована на вичерпання пулу IP-адрес шляхом масових запитів з підробленими MAC-адресами. Для захисту використовуються такі механізми, як DHCP Snooping на комутаторах, обмеження кількості MAC-адрес на порт, сегментація мережі та використання автентифікації на рівні доступу.

Переваги та недоліки використання DHCP

Основними перевагами DHCP є зменшення навантаження на адміністратора, мінімізація помилок конфігурації, централізоване управління параметрами мережі та ефективне використання IP-адресного простору. Недоліками можна вважати залежність від доступності DHCP-сервера та потенційні ризики безпеки у разі неправильної конфігурації.

DHCP є базовим і невід'ємним компонентом сучасних IP-мереж. Його використання є стандартом де-факто як у корпоративних, так і в домашніх мережах.

DHCPv6 (Dynamic Host Configuration Protocol for IPv6)

Перехід від IPv4 до IPv6 зумовив необхідність адаптації базових мережевих сервісів, зокрема механізмів автоматичної конфігурації вузлів. Одним із ключових таких сервісів є DHCPv6 − протокол динамічної конфігурації хостів для мереж IPv6. DHCPv6 забезпечує централізоване та кероване надання параметрів мережевої конфігурації клієнтам, що є особливо важливим у корпоративних, операторських і хмарних середовищах.

На відміну від IPv4, у світі IPv6 поряд із DHCPv6 активно використовується механізм Stateless Address Autoconfiguration (SLAAC). Це призводить до необхідності чіткого розуміння ролей, переваг і обмежень кожного підходу, а також їх спільного використання.

DHCPv6 призначений для автоматичного надання вузлам мережі IPv6 таких параметрів: адрес IPv6 або префіксів; адрес DNS-серверів; імен доменів пошуку; адрес NTP-серверів; інших параметрів, визначених стандартами IETF. DHCPv6 не замінює повністю механізми IPv6, а доповнює їх. У типовій мережі IPv6 маршрутизатор відіграє ключову роль, передаючи клієнтам інформацію про префікс і доступність DHCPv6 за допомогою Router Advertisement (RA) повідомлень.

Основні відмінності DHCPv6 від DHCP для IPv4

DHCPv6 має низку принципових відмінностей від DHCPv4. По-перше, DHCPv6 не використовує широкомовні повідомлення, оскільки IPv6 не підтримує broadcast. Натомість застосовується мультикаст, зокрема адреса ff02::1:2 для DHCPv6 серверів і ретрансляторів.

По-друге, у DHCPv6 ідентифікація клієнта не базується безпосередньо на MAC-адресі. Замість цього використовуються ідентифікатори DUID (DHCP Unique Identifier), що забезпечує стабільнішу ідентифікацію навіть при зміні мережевого інтерфейсу.

По-третє, DHCPv6 тісно інтегрований із механізмами IPv6 Neighbor Discovery та Router Advertisement, що визначають модель взаємодії клієнта, маршрутизатора і DHCPv6 сервера.

Архітектура DHCPv6 включає три основні компоненти: клієнт, сервер і ретранслятор. Клієнт − це вузол IPv6, який потребує конфігураційних параметрів. Сервер DHCPv6 зберігає політики адресації та параметрів і відповідає на запити клієнтів. Ретранслятор використовується в сегментованих мережах для передавання DHCPv6 повідомлень між клієнтами і сервером, якщо вони знаходяться в різних мережевих сегментах. Обмін повідомленнями відбувається поверх протоколу UDP із використанням порту 546 для клієнта та 547 для сервера.

На практиці IPv6 розрізняють три основні моделі конфігурації:

– у режимі Stateless DHCPv6 клієнт самостійно формує IPv6-адресу за допомогою SLAAC, а DHCPv6 використовується лише для отримання додаткових параметрів, таких як DNS;

– у режимі Stateful DHCPv6 сервер повністю відповідає за надання IPv6-адрес клієнтам, веде облік виданих адрес і контролює їх життєвий цикл;

– комбінований підхід передбачає одночасне використання SLAAC для адресації та DHCPv6 для сервісних параметрів, що є найпоширенішим варіантом у сучасних мережах.

Повідомлення та процес роботи DHCPv6

Робота DHCPv6 базується на обміні стандартизованими повідомленнями. Процес починається з повідомлення Solicit, яке клієнт надсилає для пошуку доступних серверів. Сервер відповідає повідомленням Advertise, інформуючи про можливість обслуговування клієнта.

Далі клієнт надсилає Request, обираючи конкретний сервер, а сервер підтверджує виділення параметрів повідомленням Reply. Для продовження терміну дії адреси використовуються повідомлення Renew і Rebind.

Час життя адрес і параметрів

DHCPv6 використовує поняття preferred lifetime і valid lifetime для керування життєвим циклом адрес. Preferred lifetime визначає період, протягом якого адреса є бажаною для нових з'єднань, тоді як valid lifetime визначає загальний час її придатності. Такий підхід дозволяє гнучко керувати перенумерацією мережі та поступовим виведенням адрес з експлуатації без порушення активних сесій.

Безпека DHCPv6 є критично важливою, оскільки зловмисний сервер може нав'язати клієнтам некоректні параметри мережі. Основними загрозами є підміна DHCPv6 сервера, атаки відмови в обслуговуванні та маніпуляції з конфігураційними параметрами. Для захисту використовуються механізми фільтрації DHCPv6 трафіку на комутаторах, функції DHCPv6 Guard, а також сегментація мережі. Додатково можливе застосування IPsec для захисту обміну повідомленнями.

SLAAC забезпечує простоту і мінімальну залежність від серверної інфраструктури, але обмежений у можливостях централізованого управління. DHCPv6, навпаки, дозволяє повністю контролювати процес конфігурації, вести облік клієнтів і реалізовувати політики безпеки. Вибір між SLAAC і DHCPv6 залежить від вимог до керованості, масштабу мережі та рівня безпеки. DHCPv6 широко використовують для централізованого управління IPv6-інфраструктурою, автоматизації адміністрування та забезпечення узгодженої роботи мережевих сервісів.

Ієрархічна структура доменів, організації, що керують адресним простором (IANA, ICANN, RIR)

Загальне керування адресним простором IP-адрес здійснює Адміністрація адресного простору Інтернет (IANA – Internet Assigned Numbers Authority), яка є підрозділом неприбуткової Інтернет-корпорації з призначення імен та адрес ICANN (Internet Corporation for Assigned Names and Numbers). IANA підпорядковуються регіональні Інтернет-реєстратори (RIR – Regional Internet Registries), яким, у свою чергу, підпорядковуються локальні Інтернет-реєстратори (LIR – Local Internet Registries) – провайдери послуг Інтернет. Регіональні Інтернет-реєстратори розподіляють IP-адреси як між кінцевими користувачами, так і між локальним Інтернет-провайдерами. Сфера впливу регіональних Інтернет-реєстраторів розповсюджується на певні регіони, а саме:

– RIPE NCC (Reseaux IP Europeens Network Coordination Centre) – Європа, Близький Схід та Центральна Азія;

– ARIN (American Registry for Internet Numbers) – Північна Америка;

– LACNIC (Latin American and Caribbean Internet Addresses Registry) – Південна Америка та басейн Карибського моря;

– APNIC (Asia-Pacific Network Information Centre) – Азійсько- Тихоокеанський регіон;

– AfriNIC (African Network Information Centre) – Африка.

Проблеми та виклики мережевої адресації: вичерпання IPv4, перехід на IPv6 та роль NAT і PAT

Стрімке зростання кількості підключених пристроїв, розвиток мобільних технологій, Інтернету речей та хмарних сервісів призвели до системної проблеми − вичерпання адресного простору IPv4.

Вичерпання пулів IPv4-адрес було офіційно зафіксоване регіональними реєстраторами, що означало неможливість подальшого виділення нових глобальних адрес у традиційному розумінні. Це стало серйозним викликом для операторів зв'язку, провайдерів

інтернет-послуг та корпоративних мереж, оскільки модель прямої унікальної адресації кожного пристрою втратила масштабованість.

У відповідь на кризу адресації було запропоновано нову версію протоколу − IPv6. Він використовує 128-бітну адресацію, що забезпечує практично невичерпний адресний простір. IPv6 не лише вирішує проблему кількості адрес, але й спрощує маршрутизацію, підтримує автоматичну конфігурацію, покращує механізми безпеки та оптимізує обробку пакетів у мережевому обладнанні. Попри очевидні переваги, перехід на IPv6 відбувається повільно через необхідність модернізації інфраструктури, програмного забезпечення та підготовки фахівців. Більшість сучасних мереж змушені працювати в умовах співіснування IPv4 та IPv6, використовуючи перехідні механізми.

Одним із ключових тимчасових рішень проблеми вичерпання IPv4 стало використання технології трансляції мережевих адрес − NAT (Network Address Translation). Основна ідея NAT полягає у відокремленні внутрішнього адресного простору мережі від глобального адресного простору Інтернету. Для цього використовуються приватні IP-адреси, визначені стандартами, які не маршрутизуються в глобальній мережі. Пристрої у внутрішній мережі можуть мати однакові приватні адреси в різних організаціях, не створюючи конфліктів.

Принцип роботи NAT базується на зміні IP-адрес у заголовках пакетів під час їх проходження через маршрутизатор або міжмережевий екран. Коли внутрішній хост ініціює з’єднання із зовнішнім ресурсом, NAT-пристрій замінює його приватну IP-адресу на публічну адресу, що належить організації або провайдеру. У таблиці трансляції зберігається відповідність між внутрішньою та зовнішньою адресами, що дозволяє коректно обробляти зворотний трафік.

Подальшим розвитком і найбільш поширеною формою NAT є PAT (Port Address Translation), або трансляція адрес із використанням портів. У цьому випадку велика кількість внутрішніх хостів може одночасно використовувати одну публічну IP-адресу, а розрізнення з’єднань відбувається за номерами транспортних портів. Кожне активне з’єднання має унікальну комбінацію зовнішньої адреси та порту, що дозволяє маршрутизатору правильно зіставляти пакети з відповідними внутрішніми пристроями.

Застосування NAT і PAT дало змогу суттєво продовжити термін використання IPv4, зменшити потребу в публічних адресах і спростити внутрішню організацію мереж. Водночас ці технології мають низку обмежень і недоліків. Вони порушують принцип наскрізної адресації, ускладнюють роботу деяких протоколів, зокрема тих, що потребують прямого встановлення з’єднання між хостами, та створюють додаткові труднощі для забезпечення прозорої безпеки. У контексті кібербезпеки NAT часто помилково сприймається як захисний механізм, хоча насправді він не замінює повноцінних засобів фільтрації та контролю доступу.

Таким чином, поява NAT була зумовлена насамперед обмеженістю IPv4-адресного простору та необхідністю зберегти працездатність Інтернету в умовах стрімкого зростання кількості підключень. Однак NAT і PAT слід розглядати лише як тимчасове рішення, яке не усуває фундаментальних обмежень застарілої архітектури. Стратегічним напрямом розвитку мереж залишається повноцінний перехід на IPv6, що відновлює початкову ідею глобальної унікальної адресації та створює основу для подальшого масштабування і розвитку цифрової інфраструктури.

Тема 3. Моделі та протоколи

Основи передавання даних. Еталонні моделі. Модель OSI: структура, рівні та функції. Модель TCP/IP: архітектура, подібності та відмінності від OSI. Порівняльний аналіз OSI та TCP/IP у практиці. Роль ISO, IEEE, ITU, IETF, ICANN. Поняття протоколу: правила, формати, послідовність дій. Інкапсуляція та декапсуляція даних. Класифікація протоколів за рівнями. Протоколи канального рівня (Ethernet, Wi-Fi). Протоколи мережевого рівня (IP, ICMP, ARP). Протоколи транспортного рівня (TCP, UDP). Прикладні протоколи (HTTP, FTP, SMTP, DNS, DHCP, DHCPv6.). Організації зі стандартизації. Інкапсуляція та доступ до даних. Механізм проходження даних через рівні моделі. Формування кадру, пакета, сегмента та повідомлення. IP-телефонія. Відстеження роботи протоколів у Wireshark. Візуалізація інкапсуляції даних у мережевих емуляторах.

Передавання даних у комп'ютерних мережах є основою сучасних інформаційних технологій і спрямоване на забезпечення логічної комунікації між системами, що можуть бути розташовані як у межах локальних мереж (LAN), так і глобальних мереж (WAN) або в Інтернеті. Функціонування мереж базується на розбитті початкового інформаційного блоку на менші одиниці (кадри, пакети, сегменти, повідомлення), передачі їх по фізичному середовищу, маршрутизації між вузлами та відновленні початкового об'єму даних у місці призначення. Цей процес опирається на множину еталонних моделей і протоколів, що визначають правила формування, адресації, маршрутизації та обробки даних.

Еталонні моделі передачі даних

Модель OSI [10] є класичною еталонною моделлю взаємодії відкритих систем та описує процес мережевого взаємодії як послідовність функціональних рівнів. Вона була розроблена Міжнародною організацією зі стандартизації (ISO) у 1978 р., щоб уніфікувати підходи до побудови мережевих протоколів та взаємодії різних систем незалежно від виробника і технології. Кожен з семи рівнів цієї моделі має чітко визначені функції:

– рівень 1 − фізичний: відповідає за передачу бітів через фізичні середовища (кабелі, радіохвилі).

– рівень 2 − канальний: забезпечує організацію доступу до середовища і передачу кадрів між безпосередньо з'єднаними вузлами.

– рівень 3 − мережевий: реалізує логічну адресацію та маршрутизацію пакетів через мережу.

– рівень 4 − транспортний: забезпечує надійне передавання сегментів або датаграм між кінцевими вузлами.

– рівні 5-7 − сеансовий, представлення та прикладний: відповідають за встановлення/закриття сеансів, представлення даних та надання сервісів для прикладних програм.

Модель OSI має важливе концептуальне значення, хоча безпосередньо не реалізується у сучасних мережах як повний стек протоколів. Проте вона лишається основою для розуміння структуризації мережевих функцій.

Модель TCP/IP є практичною архітектурою, на якій побудований Інтернет і більшість сучасних мереж. Ця модель виникла раніше за OSI та була розроблена як результат практичних потреб створення глобальної мережі з відкритою взаємодією пристроїв. На відміну від OSI, TCP/IP має чотири логічні рівні:

– рівень доступу до мережі (Network Access / Link Layer): включає фізичні та канальні протоколи, наприклад Ethernet, Wi-Fi − компоненти, що відповідають за передачу кадрів в межах однієї фізичної мережі;

– Internet Layer (мережевий): відповідає за логічну адресацію і маршрутизацію через протокол IP та пов'язані механізми, такі як ICMP і ARP;

– Transport Layer (транспортний): забезпечує передачу даних між кінцевими точками за допомогою TCP (надійне передавання) або UDP (ненадійне, але швидке);

– Application Layer (прикладний): включає сервіси, що надають прикладні функції, наприклад HTTP, FTP, SMTP, DNS та DHCP.

Ця архітектура віддзеркалює набір реальних протоколів, які взаємодіють для забезпечення передачі даних у глобальних мережах, і є основою для виконання мережевих сервісів.

Порівняльний аналіз OSI та TCP/IP у практиці

Порівняння моделей OSI та TCP/IP показує, що OSI слугує концептуальним еталоном, що розділяє мережеві функції на семи рівнях, тоді як TCP/IP − це практичний стек протоколів, що реалізує передачу даних у реальних мережах з меншим числом абстрактних рівнів та злитими функціями. Наприклад, у TCP/IP фізичний і канальний рівні OSI об'єднані у рівень доступу до мережі, а OSI-рівні представлення й сеансу функціонально входять у прикладний рівень TCP/IP. Попри ці відмінності, для аналізу мережевих проблем і проектування мереж часто користуються моделлю OSI як логічним каркасом, а TCP/IP − для практичної реалізації передачі даних.

У мережевих технологіях протокол − це формалізований набір прав, форматів, механізмів взаємодії і процедур, що визначають, як дані повинні бути передані між двома чи більше кінцевими точками у мережі. Протокол визначає:

Формати повідомлень (структури заголовків і полів) і синтаксис передачі;

Правила послідовності дій (коли і як ініціюються, підтримуються і завершуються обміни);

Механізми обробки помилок та контроль цілісності.

Ці визначення реалізуються на кожному рівні моделі OSI або TCP/IP, забезпечуючи чітке розмежування відповідальності між функціональними компонентами.

Протоколи класифікують за рівнями моделі OSI/TCP-IP:

Канальний рівень [10]: Ethernet, Wi-Fi − визначають формування і структуру кадрів у локальних мережах, доступ до фізичного середовища та MAC-адресацію.

Мережевий рівень [10]: IP − забезпечує логічну адресацію та маршрутизацію пакетів; ICMP − служить для контролю та обміну службовою інформацією про стан мережі; ARP − визначає відповідність між IP-адресою та MAC-адресою.

Транспортний рівень: TCP − забезпечує надійне передавання з контролем потоку і помилок; UDP − легкий, ненадійний протокол для швидких передач, де гарантована доставка не критична.

Прикладний рівень: HTTP, FTP, SMTP, DNS, DHCP (і його IPv6-версія DHCPv6) − реалізують прикладні сервіси користувача: веб-обмін, передавання файлів, електронна пошта, розв'язання доменних імен та автоматичне налаштування параметрів мережевих клієнтів.

Роль ISO, IEEE, ITU, IETF, ICANN у розвитку мережевих технологій

Розвиток комп'ютерних мереж і глобальних телекомунікацій був би неможливий без діяльності міжнародних організацій, які займаються розробленням стандартів, регулюванням і координацією технічних рішень. Найважливішу роль у цьому процесі відіграють ISO, IEEE, ITU, IETF та ICANN.

ISO (International Organization for Standardization) є міжнародною організацією зі стандартизації, що розробляє загальні стандарти для різних галузей, зокрема інформаційних технологій. У сфері комп'ютерних мереж ISO відома насамперед створенням еталонної моделі OSI, яка стала теоретичною основою для розуміння принципів мережевої взаємодії. Стандарти ISO сприяють сумісності обладнання та програмного забезпечення різних виробників і забезпечують єдиний підхід до проєктування систем.

IEEE (Institute of Electrical and Electronics Engineers) зосереджується на технічних стандартах у галузі електроніки, електротехніки та комп'ютерних мереж. Саме IEEE розробляє стандарти сімейства 802, до яких належать Ethernet (IEEE 802.3) та бездротові мережі Wi-Fi (IEEE 802.11). Ці стандарти визначають роботу фізичного та канального рівнів і є основою більшості сучасних локальних мереж.

ITU (International Telecommunication Union) є спеціалізованою установою ООН, що координує глобальні телекомунікації. Вона розробляє міжнародні рекомендації для телефонного зв'язку, передачі даних, радіозв'язку та супутникових систем. ITU також

відповідає за розподіл радіочастотного спектра і орбітальних позицій супутників, що має ключове значення для стабільної роботи світових систем зв'язку.

IETF (Internet Engineering Task Force) відіграє провідну роль у розвитку Інтернету. Це відкрита спільнота фахівців, яка розробляє та публікує стандарти у вигляді документів RFC (Request for Comments). Саме IETF створила ключові інтернет-протоколи, зокрема IP, TCP, UDP, HTTP, SMTP і DNS. Її діяльність забезпечує еволюцію Інтернету, зберігаючи сумісність і відкритість мережі.

ICANN (Internet Corporation for Assigned Names and Numbers) відповідає за координацію унікальних ідентифікаторів Інтернету. До її повноважень належать управління доменними іменами, IP-адресним простором і кореневими DNS-серверами. Завдяки ICANN забезпечується унікальність адрес і доменів, що є необхідною умовою стабільної та безперервної роботи глобальної мережі.

Таким чином, ISO, IEEE, ITU, IETF та ICANN виконують взаємодоповнювальні функції, формуючи технічну, організаційну та регуляторну основу сучасних комп'ютерних мереж і Інтернету.

Поняття протоколу в комп'ютерних мережах

Функціонування комп'ютерних мереж ґрунтується на чітко визначених правилах взаємодії між пристроями, незалежно від їх апаратної платформи, операційної системи чи виробника. Сукупність таких правил у теорії та практиці мережевих технологій отримала назву протоколу. Мережевий протокол − це формалізований опис правил, форматів і процедур, які визначають спосіб обміну даними між двома або більше вузлами мережі.

Протокол задає не лише форму передавання інформації, а й логіку взаємодії: коли і за яких умов відбувається передавання, як ініціюється з'єднання, яким чином підтверджується успішна доставка даних, як здійснюється контроль помилок і відновлення після збоїв. Без узгоджених протоколів взаємодія в мережі була б неможливою, оскільки кожен пристрій інтерпретував би дані по-своєму.

У структурі будь-якого протоколу можна виділити три базові складові: правила, формати та послідовність дій. Правила визначають допустиму поведінку учасників обміну даними, включно з ролями сторін, умовами початку й завершення сеансу зв'язку та реакцією на помилки. Формати описують структуру повідомлень, зокрема порядок розташування полів заголовка, довжину даних, способи кодування адресної та службової інформації. Послідовність дій регламентує часову логіку обміну, тобто визначає, які повідомлення і в якій черговості мають бути надіслані для коректної взаємодії.

Таким чином, протокол є основою сумісності мережевих пристроїв і програмних систем, а також гарантом надійності та передбачуваності передавання інформації.

Інкапсуляція та декапсуляція даних

Процес передавання даних у мережі відбувається поетапно і базується на принципі багаторівневої обробки інформації. Центральним механізмом цього процесу є інкапсуляція даних. Інкапсуляція полягає в послідовному додаванні службової інформації до користувацьких даних під час їх проходження через рівні мережевої моделі від прикладного рівня до фізичного.

На кожному рівні мережевої архітектури до даних додається власний заголовок, а в окремих випадках − і трейлер. Ці службові поля містять інформацію, необхідну для коректної доставки повідомлення, зокрема адреси відправника й отримувача, ідентифікатори протоколів, контрольні суми, номери портів або послідовності сегментів. У результаті первинні дані перетворюються на складну структуру, яка послідовно набуває форми сегмента, пакета, кадру або бітового потоку.

Протилежним процесом є декапсуляція, яка відбувається на стороні отримувача. Під час декапсуляції дані рухаються знизу вгору по рівнях мережевої моделі. Кожен рівень аналізує та обробляє відповідний заголовок, після чого видаляє його і передає корисне навантаження на вищий рівень. У кінцевому підсумку прикладна програма отримує дані в тому вигляді, в якому вони були сформовані відправником.

Інкапсуляція та декапсуляція забезпечують модульність мережевої архітектури, що дозволяє незалежно розвивати та вдосконалювати окремі протоколи без порушення роботи всієї системи.

Класифікація протоколів за рівнями мережевої моделі

Для систематизації протоколів і спрощення проектування мереж застосовується рівнева організація, найвідомішими реалізаціями якої є модель OSI та модель TCP/IP. Класифікація протоколів за рівнями дає змогу чітко визначити їх функціональне призначення та область відповідальності.

На прикладному рівні зосереджені протоколи, безпосередньо орієнтовані на взаємодію з користувачем або прикладними програмами. Вони забезпечують доступ до мережевих сервісів, таких як передавання файлів, електронна пошта, веб-доступ або служби каталогів. Ці протоколи не займаються доставкою пакетів у фізичному сенсі, а формують логіку прикладної взаємодії.

Транспортний рівень відповідає за наскрізне передавання даних між вузлами, контроль цілісності, управління потоком та, за потреби, надійність доставки. Протоколи цього рівня працюють із логічними з'єднаннями та забезпечують мультиплексування даних між кількома прикладними процесами.

На мережевому рівні функціонують протоколи, що забезпечують логічну адресацію та маршрутизацію даних між різними мережами. Саме на цьому рівні приймаються рішення щодо вибору маршруту, обробляються таблиці маршрутизації та реалізується доставка пакетів до кінцевого вузла через проміжні маршрутизатори.

Канальний рівень об'єднує протоколи, які регламентують передавання даних у межах одного фізичного сегмента мережі. Вони відповідають за формування кадрів, фізичну адресацію, виявлення помилок та управління доступом до середовища передавання.

Нарешті, фізичний рівень визначає електричні, оптичні або радіотехнічні характеристики передавання сигналів. Хоча формально він не оперує протоколами у класичному розумінні, саме цей рівень забезпечує фізичну можливість передавання бітів у середовищі зв'язку.

Узгоджена робота протоколів усіх рівнів формує єдину систему передавання даних, у якій кожен рівень виконує строго визначену функцію, не дублюючи і не підміняючи інші.

Мережеві протоколи є фундаментом сучасних комп'ютерних мереж, забезпечуючи стандартизовану, надійну та масштабовану взаємодію між пристроями. Поняття протоколу охоплює правила, формати та послідовність дій, необхідні для коректного обміну даними. Процеси інкапсуляції та декапсуляції реалізують багаторівневу обробку інформації, що дозволяє абстрагувати складність мережевих технологій. Класифікація протоколів за рівнями дає змогу структурувати мережеві рішення та ефективно проектувати, впроваджувати й аналізувати комп'ютерні мережі різного масштабу та призначення.

Протоколи канального рівня (Ethernet, Wi-Fi)

Канальний рівень [10] є другим рівнем еталонної семирівневої моделі OSI та відіграє ключову роль у забезпеченні надійної передачі даних між безпосередньо з'єднаними вузлами мережі. Саме на цьому рівні відбувається формування кадрів, фізична адресація пристроїв, виявлення помилок передавання та керування доступом до спільного середовища передавання даних. Протоколи канального рівня тісно взаємодіють із фізичним рівнем, використовуючи його для передачі бітів, але водночас забезпечують логічну структуру передавання інформації у вигляді кадрів.

Канальний рівень традиційно поділяється на два підрівні: підрівень логічного керування каналом (Logical Link Control, LLC) та підрівень керування доступом до середовища (Media Access Control, MAC). Підрівень LLC відповідає за ідентифікацію протоколів мережевого рівня та логічний контроль з'єднання, тоді як підрівень MAC реалізує механізми адресації та доступу до фізичного середовища передавання. Найпоширенішими технологіями канального рівня в локальних мережах є Ethernet та Wi-Fi, стандартизовані відповідно організацією IEEE у серіях стандартів IEEE 802.3 та IEEE 802.11.

Ethernet є базовою технологією дротових локальних обчислювальних мереж. Вона була розроблена для забезпечення простого, масштабованого та ефективного передавання

даних у середовищі спільного доступу. Класична модель Ethernet передбачала використання методу множинного доступу з виявленням колізій (CSMA/CD), за якого вузол перед початком передавання перевіряє, чи є середовище вільним, а у випадку одночасної передачі кількох станцій відбувається колізія, що виявляється та обробляється повторною передачею кадру. З розвитком комутованих мереж і повнодуплексного режиму роботи роль механізму CSMA/CD значно зменшилася, однак він залишається важливою частиною теоретичних основ Ethernet.

Кадр Ethernet має чітко визначену структуру, яка включає преамбулу та стартовий обмежувач кадру, MAC-адресу призначення, MAC-адресу джерела, поле типу або довжини, поле корисних даних та контрольну суму кадру (Frame Check Sequence, FCS). MAC-адреси в Ethernet є унікальними 48-бітними ідентифікаторами, що присвоюються мережевим інтерфейсам і забезпечують однозначну ідентифікацію вузлів у межах локального сегмента. Поле FCS використовується для виявлення помилок за допомогою циклічного надлишкового коду (CRC), що дозволяє приймальній стороні визначити факт спотворення даних під час передавання.

Ethernet підтримує різні швидкості та типи фізичного середовища, починаючи від класичних 10 Мбіт/с і завершуючи сучасними стандартами 10, 40, 100 Гбіт/с і більше. У практиці сучасних комп'ютерних мереж Ethernet є основою корпоративних, датацентрових та провайдерських інфраструктур завдяки своїй простоті, надійності та широкій підтримці обладнання.

На відміну від Ethernet, технологія Wi-Fi забезпечує бездротову передачу даних у локальних мережах і ґрунтується на використанні радіоканалу як середовища передавання. Wi-Fi також належить до стандартів сімейства IEEE 802, зокрема до серії IEEE 802.11, яка визначає фізичний і канальний рівні для бездротових мереж. Спільний характер радіоефіру та неможливість прямого виявлення колізій зумовили використання іншого механізму доступу до середовища − множинного доступу з уникненням колізій (CSMA/CA).

У Wi-Fi перед початком передачі станція не лише перевіряє зайнятість каналу, а й використовує випадкову затримку та механізми підтвердження доставки кадру. Кожен успішно прийнятий кадр підтверджується спеціальним керуючим повідомленням ACK, що підвищує надійність передавання, але водночас збільшує накладні витрати протоколу. За відсутності підтвердження кадр вважається втраченим і передається повторно.

Кадр Wi-Fi має складнішу структуру порівняно з Ethernet і може містити до чотирьох MAC-адрес, що пов'язано з особливостями бездротової архітектури, зокрема з використанням точок доступу та режимів ретрансляції. У протоколі передбачені різні типи кадрів: кадри даних, керуючі кадри та кадри управління, які використовуються для встановлення з'єднання, аутентифікації, асоціації клієнтів і керування енергоспоживанням.

Особливу увагу в технології Wi-Fi приділено питанням безпеки. На канальному рівні реалізуються механізми шифрування та автентифікації, зокрема стандарти WPA2 та WPA3, які забезпечують конфіденційність і цілісність переданих даних у бездротовому середовищі. Це є принциповою відмінністю від класичного Ethernet, де питання безпеки зазвичай вирішуються на вищих рівнях або за допомогою додаткових технологій, таких як VLAN, MAC-фільтрація чи мережевий контроль доступу.

Порівнюючи Ethernet і Wi-Fi, слід зазначити, що обидві технології реалізують однакові базові функції канального рівня, але адаптовані до різних фізичних середовищ. Ethernet забезпечує вищу стабільність, менші затримки та більшу пропускну здатність, тоді як Wi-Fi надає мобільність і гнучкість підключення за рахунок бездротового доступу. У сучасних мережах ці технології не конкурують, а взаємодоповнюють одна одну, формуючи єдину інфраструктуру доступу до мережевих ресурсів.

Таким чином, протоколи канального рівня Ethernet і Wi-Fi є фундаментом локальних комп'ютерних мереж. Їх розуміння є необхідним для ефективного проєктування, адміністрування та захисту мережевих систем, а також для подальшого вивчення протоколів вищих рівнів моделі OSI та сучасних мережевих технологій.

Протоколи мережевого рівня (IP, ICMP, ARP) та транспортного рівня (TCP, UDP)

Функціонування комп'ютерних мереж ґрунтується на ієрархічній організації протоколів, кожен з яких виконує чітко визначені завдання. У моделях OSI та TCP/IP ключову роль у передаванні даних між вузлами відіграють мережевий і транспортний рівні, які забезпечують адресацію, маршрутизацію, доставку та контроль передавання інформації.

Мережевий рівень та його протоколи

Мережевий рівень [10] відповідає за логічну адресацію вузлів, вибір маршруту передавання даних і доставку пакетів між різними мережами. Центральним протоколом цього рівня є Internet Protocol (IP), який працює разом із допоміжними протоколами ICMP та ARP.

Internet Protocol (IP) є базовим протоколом Інтернету та реалізує принцип best effort delivery, тобто не гарантує доставку пакетів, їх порядок чи відсутність дублювання. IP виконує фрагментацію даних, інкапсуляцію сегментів транспортного рівня в пакети та забезпечує їх маршрутизацію. Існують дві основні версії протоколу: IPv4, що використовує 32-бітну адресу, та IPv6, який застосовує 128-бітну адресацію й усуває низку обмежень IPv4, зокрема дефіцит адресного простору та залежність від NAT.

IP не здійснює контроль помилок на рівні доставки, однак забезпечує базову перевірку цілісності заголовка пакета. Вся відповідальність за надійність передавання покладається на транспортний рівень.

Internet Control Message Protocol (ICMP) використовується для передавання службових і діагностичних повідомлень. ICMP не призначений для транспортування прикладних даних, але відіграє важливу роль у керуванні мережею та виявленні помилок. За його допомогою вузли повідомляють про недоступність мережі або хоста, перевищення часу життя пакета (TTL), проблеми маршрутизації. Практичне значення ICMP проявляється в таких утилітах, як ping і traceroute, які застосовуються для діагностики стану мережі.

Address Resolution Protocol (ARP) забезпечує взаємозв'язок між логічною IP-адресою та фізичною MAC-адресою у локальній мережі. Оскільки передавання кадрів у канальному рівні відбувається за MAC-адресами, вузол повинен знати фізичну адресу отримувача. ARP працює шляхом широкомовного запиту, у відповідь на який вузол із відповідною IP-адресою надсилає свою MAC-адресу. Отримані відповідності зберігаються в ARP-кеші. У контексті кібербезпеки ARP має критичне значення, оскільки є вразливим до атак типу ARP-spoofing або ARP-poisoning.

Транспортний рівень та його протоколи

Транспортний рівень забезпечує логічну взаємодію між прикладними процесами на різних вузлах мережі. Основними протоколами цього рівня є Transmission Control Protocol (TCP) та User Datagram Protocol (UDP), які суттєво відрізняються за принципами роботи.

Transmission Control Protocol (TCP) є орієнтованим на з'єднання протоколом, який гарантує надійне та впорядковане доставлення даних. Перед початком обміну TCP встановлює з'єднання за допомогою триетапного рукостискання (three-way handshake), що забезпечує синхронізацію між відправником і отримувачем. TCP виконує контроль цілісності даних, керування потоком, повторну передачу втрачених сегментів та адаптацію швидкості передавання відповідно до завантаженості мережі.

Завдяки цим властивостям TCP широко використовується у сервісах, де критично важлива коректність і повнота передавання даних, зокрема у веб-з'єднаннях (HTTP/HTTPS), електронній пошті (SMTP, IMAP, POP3) та передаванні файлів (FTP).

User Datagram Protocol (UDP) є протоколом без встановлення з'єднання, який не гарантує доставку, порядок або відсутність дублювання пакетів. Основною перевагою UDP є мінімальні накладні витрати та висока швидкість передавання. Протокол не виконує керування потоком чи повторної передачі даних, що робить його ефективним для застосувань реального часу.

UDP широко використовується в потоковому відео та аудіо, онлайн-іграх, системах VoIP, а також у службових протоколах, таких як DNS і DHCP. Надійність у таких системах зазвичай реалізується на прикладному рівні.

Порівняльний аналіз та практичне значення

Мережевий і транспортний рівні взаємодіють між собою, утворюючи основу міжмережевої комунікації. IP забезпечує доставку пакетів між мережами, ICMP − діагностику та керування помилками, ARP − зв'язок між логічною та фізичною адресацією. TCP і UDP, у свою чергу, надають прикладним програмам різні моделі обміну даними, дозволяючи обирати між надійністю та швидкодією.

Прикладні протоколи комп'ютерних мереж

Прикладний рівень є верхнім рівнем як у моделі OSI, так і в архітектурі TCP/IP. Його основне призначення полягає у забезпеченні безпосередньої взаємодії мережевих сервісів з користувацькими застосунками. Саме на цьому рівні реалізуються протоколи, які визначають правила обміну даними між програмами, що працюють на різних вузлах мережі.

Протоколи прикладного рівня [10] не займаються маршрутизацією, фрагментацією або фізичною передачею сигналів. Вони використовують сервіси транспортного рівня (TCP або UDP) для доставки даних і зосереджуються на форматі повідомлень, логіці обміну та семантиці запитів і відповідей. До найпоширеніших прикладних протоколів належать HTTP, FTP, SMTP, DNS, DHCP та DHCPv6, які забезпечують функціонування веб-сервісів, передавання файлів, електронної пошти та автоматичну конфігурацію мережевих параметрів.

HTTP (HyperText Transfer Protocol) − це протокол прикладного рівня, призначений для передавання гіпертекстових документів у Всесвітній павутині. Він лежить в основі роботи веб-браузерів і веб-серверів та забезпечує доступ до веб-сторінок, зображень, відео, API-сервісів і хмарних застосунків.

HTTP є протоколом типу «клієнт–сервер». Клієнт, зазвичай веб-браузер, ініціює з'єднання та надсилає HTTP-запит, а сервер обробляє його і повертає HTTP-відповідь. Протокол є безстанним, тобто сервер не зберігає інформацію про попередні запити клієнта, якщо це не реалізовано додатковими механізмами, такими як cookies або сесії.

Основними методами HTTP є GET, POST, PUT, DELETE, HEAD та OPTIONS. Вони визначають тип дії, яку клієнт хоче виконати над ресурсом. Для передавання даних HTTP зазвичай використовує TCP-порт 80, а його захищена версія HTTPS, що працює з використанням TLS, − порт 443.

FTP (File Transfer Protocol) − це прикладний протокол, призначений для передавання файлів між клієнтом і сервером у мережі TCP/IP. Він широко використовувався для обміну файлами в локальних і глобальних мережах, хоча в сучасних системах часто замінюється більш безпечними рішеннями.

Особливістю FTP є використання двох з'єднань: керуючого та з'єднання для передавання даних. Керуюче з'єднання встановлюється через TCP-порт 21 і використовується для передавання команд, тоді як дані передаються через окреме з'єднання, параметри якого залежать від режиму роботи − активного або пасивного.

FTP не забезпечує шифрування облікових даних та передаваних даних, що є його суттєвим недоліком з точки зору безпеки. Для усунення цієї проблеми були розроблені розширення FTPS та альтернативний протокол SFTP, який працює поверх SSH.

SMTP (Simple Mail Transfer Protocol) − це прикладний протокол, призначений для передавання електронної пошти між поштовими серверами. Він забезпечує пересилання повідомлень від клієнта до сервера, а також між серверами електронної пошти.

SMTP використовує модель «store-and-forward», за якої повідомлення тимчасово зберігається на проміжних серверах до моменту доставки кінцевому адресату. Протокол працює поверх TCP і за замовчуванням використовує порт 25, а також порти 587 та 465 для захищеної передачі з використанням TLS.

SMTP відповідає лише за надсилання пошти. Для отримання повідомлень використовуються інші прикладні протоколи, зокрема POP3 та IMAP. У сучасних системах SMTP тісно інтегрується з механізмами автентифікації, шифрування та захисту від спаму.

DNS (Domain Name System) − це розподілена система імен, яка забезпечує перетворення символьних доменних імен у IP-адреси та навпаки. DNS є критично важливим

елементом функціонування Інтернету, оскільки дозволяє користувачам звертатися до ресурсів за зрозумілими іменами замість числових адрес.

DNS організований у вигляді ієрархічної розподіленої бази даних, що складається з кореневих серверів, серверів доменів верхнього рівня та авторитетних серверів доменів. Запити DNS зазвичай передаються з використанням UDP через порт 53, а у випадках великих відповідей або зонних трансферів використовується TCP.

DNS підтримує різні типи ресурсних записів, серед яких A, AAAA, MX, CNAME, NS та TXT. Кожен із них виконує конкретну функцію, наприклад, визначення IP-адреси вузла або поштового сервера домену.

DHCP (Dynamic Host Configuration Protocol) − це прикладний протокол, який використовується для автоматичної конфігурації мережевих параметрів вузлів у IP-мережах. Він дозволяє динамічно призначати IP-адреси, маски підмереж, шлюзи за замовчуванням, DNS-сервери та інші параметри без ручного втручання адміністратора.

DHCP працює за моделлю клієнт–сервер і використовує UDP-порти 67 для сервера та 68 для клієнта. Процес отримання адреси складається з чотирьох основних етапів: виявлення сервера, пропозиції адреси, запиту та підтвердження. Така послідовність відома як DORA-процес.

Завдяки DHCP значно спрощується адміністрування мереж, зменшується кількість помилок конфігурації та забезпечується ефективне використання адресного простору IPv4.

DHCPv6 є розвитком протоколу DHCP, адаптованим для роботи в мережах IPv6. Він призначений для автоматичної конфігурації параметрів IPv6-вузлів, зокрема адрес, префіксів, DNS-серверів та інших мережевих опцій.

На відміну від IPv4, у середовищі IPv6 можливе поєднання DHCPv6 з механізмом SLAAC, який дозволяє вузлам самостійно формувати адресу на основі оголошень маршрутизатора. DHCPv6 може працювати як у режимі повного керування адресацією, так і в режимі надання лише додаткових параметрів конфігурації.

Протокол використовує UDP-порти 546 для клієнта та 547 для сервера. DHCPv6 відіграє важливу роль у сучасних корпоративних і провайдерських мережах, де впроваджується IPv6-адресація.

Прикладні протоколи є ключовим елементом функціонування комп’ютерних мереж, оскільки саме вони забезпечують реалізацію мережевих сервісів, з якими безпосередньо взаємодіє користувач. HTTP, FTP, SMTP, DNS, DHCP та DHCPv6 охоплюють основні сценарії використання мереж − від доступу до веб-ресурсів і передавання файлів до електронної пошти та автоматичної конфігурації вузлів.

Організації зі стандартизації мережевих технологій

Ефективна робота глобальних мереж і взаємодія обладнання різних виробників можливі завдяки діяльності міжнародних організацій, що займаються узгодженням і стандартизацією мережевих протоколів і технологій.

International Organization for Standardization (ISO): добровільна міжнародна організація, що розробляє міжнародні стандарти, включно з OSI Reference Model, і забезпечує загальні правила та рекомендації для комунікаційних протоколів.

Institute of Electrical and Electronics Engineers (IEEE): професійне товариство, що розробляє технічні стандарти у галузі електротехніки та обчислювальної техніки. Комітет IEEE 802 стандартизує поширені локальні мережеві технології, зокрема Ethernet (802.3) та Wi-Fi (802.11).

International Telecommunication Union (ITU-T): сектор стандартів телекомунікацій, що видає рекомендації щодо цифрових телекомунікацій та мережевої взаємодії, включно з компонентами фізичного та канального рівнів мереж.

Internet Engineering Task Force (IETF): відкрита міжнародна спільнота інженерів і дослідників, яка розробляє стандарти для Інтернет-протоколів, описані у формі RFC (Request For Comments). На IETF лежить відповідальність за технічні специфікації основних протоколів стеку TCP/IP.

Internet Corporation for Assigned Names and Numbers (ICANN): неурядова організація, що координує розподіл IP-адрес, управління DNS (система доменних імен) та роботу кореневих серверів, що є критично важливими для функціонування глобального Інтернету.

Ці організації тісно взаємодіють між собою та підтримують розроблення відкритих стандартів, що забезпечують інтероперабельність мереж та пристроїв по всьому світу.

Механізм інкапсуляції − це процес, у якому кожен рівень моделі додає власний заголовок (інколи трейлер) до даних, що надходять із вищого рівня, формуючи відповідний PDU (Protocol Data Unit) − сегменти, пакети, кадри. У зворотному порядку декапсуляція відбувається при прийомі даних, коли кожен рівень видаляє свою обгортку та передає корисну інформацію вище.

Кадр (Frame) формується на канальному рівні, пакет (Packet) − на мережевому, сегмент (Segment) − на транспортному, а повідомлення (Message) − на прикладному рівні. Цей механізм дозволяє ізольовано обробляти мережеві функції на кожному рівні та забезпечує модульність дизайну мереж.

Механізм проходження даних через рівні моделі

Коли прикладна програма генерує дані (наприклад HTTP-запит), вони передаються вниз по стеку протоколів. На кожному рівні додається відповідний заголовок, що містить службову інформацію (адреси, номери портів, контрольні суми). Далі сформовані кадри передаються через фізичне середовище до приймача, де процес повторюється у зворотному порядку − заголовки видаляються, і корисна інформація передається вище. Цей послідовний «спуск» і «підйом» через рівні і є суттю логічної передачі даних.

Практичні аспекти: IP-телефонія, Wireshark, візуалізація інкапсуляції

IP-телефонія (VoIP) − приклад мережевого сервісу, де аудіо передається як цифрові пакети через IP-мережі, використовуючи транспортні протоколи (часто UDP для зменшення затримок), і прикладні протоколи сигналізації (наприклад SIP). Надійність і якість у VoIP залежать від правильного маршрутування, контролю потоку та мінімізації втрат пакетів.

Wireshark [11] − це мережевий аналізатор, що дозволяє «перехоплювати» та візуально досліджувати трафік мережі. Використовуючи відстеження протоколів, можна бачити, як інкапсульовані дані проходять через рівні, а також аналізувати заголовки кадрів, пакетів і сегментів у реальному часі.

Тема 4. Принципи комутації та маршрутизації

Роль комутації та маршрутизації в мережах різних рівнів. Історичний розвиток технологій передавання даних. Основні поняття комутації. Методи комутації. Організація роботи комутаторів другого рівня (Layer 2 Switch). Забезпечення відмовостійкості (Spanning Tree Protocol).

Принципи маршрутизації. Визначення маршрутизації та її відмінність від комутації. Таблиці маршрутизації, метрики та алгоритми вибору шляху. Статична маршрутизація та її застосування. Динамічна маршрутизація: основні протоколи (RIP, OSPF, EIGRP, BGP). Маршрутизатори багаторівневі (Layer 3 Switch) та інтеграція з комутаторами. Порівняння комутації та маршрутизації: спільні риси та відмінності. Конфігурація комутатора та маршрутизатора у Cisco Packet Tracer. Значення принципів комутації та маршрутизації для розвитку сучасних мереж. Перспективи вдосконалення технологій (SDN, централізоване управління трафіком).

Комутація і сегментація мережі

Хоча для створення корпоративної мережі використовуються як комутатори, так і маршрутизатори, архітектура більшості корпоративних мереж у значній мірі ґрунтується на комутаторах. Вартість комутаторів з розрахунку на порт, нижча, ніж у маршрутизаторів, і вони забезпечують швидке пересилання кадрів зі швидкістю передачі даних по кабелю.

Комутатор – універсальний пристрій 2-го рівня, який використовується для з'єднання декількох вузлів. У більш складному варіанті комутатор підключається до одного чи декількох комутаторів для створення, контролю та обслуговування резервних каналів і з'єднань VLAN. Комутатор однаково обробляє всі типи трафіку, незалежно від їхнього призначення.

Комутатор передає трафік у відповідності до MAC-адрес. Кожен комутатор тримає таблицю MAC-адрес у високопродуктивній пам'яті, що називається асоціативною пам'яттю (CAM). Комутатор заново створює таблицю при кожній активації, використовуючи MAC-адреси джерела вхідних кадрів і номера портів, через які вони отримані. Комутатор видаляє запис з таблиці MAC-адрес, якщо вони не використовуються протягом визначеного періоду часу. Цей період називається таймером старіння (aging timer). Як тільки одноадресний кадр прибуває на порт, комутатор знаходить MAC-адресу джерела в кадрі. Потім він виконує пошук по таблиці MAC-адрес і знаходить запис, що відповідає адресі. Якщо MAC-адреса відсутня в таблиці, комутатор додає MAC-адресу і номер порту та активує таймер старіння. Якщо MAC-адреса джерела вже існує, комутатор скидає таймер старіння. Після цього комутатор шукає MAC-адресу призначення в таблиці MAC-адрес. Якщо запис існує, комутатор пересилає кадр на порт із відповідним номером. Якщо запису не існує, комутатор виконує лавинну маршрутизацію (floods) кадру через всі порти, крім порту, на якому він був прийнятий [11].

У корпоративному середовищі висока доступність, швидкість та смуга пропускання мережі мають першорядне значення. Розмір доменів широкомовного розсилання і доменів колізій впливає на потоки трафіку. Як правило, великі домени широкомовного розсилання і домени колізій погіршують ці критично важливі показники.

Якщо комутатор отримує широкомовний кадр, він розсилає його з усіх активних інтерфейсів так, як кадр із невідомою MAC-адресою призначення. Усі пристрої, що отримують широкомовне розсилання, складають домен широкомовного розсилання. При збільшенні числа з'єднаних комутаторів, розмір домену широкомовного розсилання також збільшується.

Домени колізій створюють аналогічну проблему. Чим більше пристроїв входить у домен колізій, тим частіше вони виникають. Комутатори використовують функцію мікросегментації, щоб зменшити розмір домену колізій до одного порту комутатора. Коли вузол підключається до порту комутатора, створюється виділене підключення. Коли два з'єднаних вузли взаємодіють один з одним, комутатор звертається до таблиці комутації і створює віртуальне підключення (мікросегмент) між портами.

Комутатор підтримує віртуальний канал (VC) до припинення сеансу. Кілька віртуальних каналів можуть бути активні одночасно. Мікросегментація покращує коефіцієнт використання смуги пропускання за рахунок зменшення кількості колізій і підтримки декількох паралельних підключень.

Комутатори можуть підтримувати симетричну та асиметричну комутацію. Комутатори, усі порти яких працюють на однаковій швидкості, називаються симетричними. Однак багато комутаторів мають два чи більше високошвидкісних порти. Ці високошвидкісні порти (порти для каскадування) використовуються для підключення до зон з більш високими вимогами до смуги пропускання. Сфери застосування таких портів:

– підключення до інших комутаторів;

– канали зв'язку з серверами;

– підключення до інших мереж.

Для з'єднання портів, що працюють на різних швидкостях, використовується асиметрична комутація. При необхідності комутатор зберігає інформацію в пам'яті, щоб створити буфер між портами з різними швидкостями. Асиметричні комутатори широко поширені в корпоративних середовищах.

Багаторівнева комутація

Традиційно мережі складалися з окремих пристроїв 2-го і 3-го рівнів. Кожен пристрій використовував різні методи обробки і пересилання трафіку.

Рівень 2. Комутатори рівня 2 є апаратними. Вони пересилають трафік зі швидкістю, що відповідає швидкості передачі середовища, використовуючи внутрішні схеми, що фізично з'єднують кожен порт з усіма іншими портами. Процес пересилання використовує MAC-адресу і наявність MAC-адреси призначення в таблиці MAC-адрес. Комутатор 2-го рівня пересилає трафік тільки всередині одного мережевого сегменту або підмережі.

Рівень 3. Маршрутизатори є програмними пристроями і використовують мікропроцесори для маршрутизації на основі IP-адрес. Маршрутизація 3-го рівня забезпечує пересилання трафіку між різними мережами та підмережами. Коли пакет приймається на інтерфейсі маршрутизатора, він використовує програмне забезпечення для пошуку IP-адреси призначення та вибору оптимального шляху до мережі призначення. Потім маршрутизатор передає пакет на потрібний вихідний інтерфейс. Багаторівневий комутатор поєднує функції комутатора 2-го рівня і маршрутизатора 3-го рівня. Комутація рівня 3 виконується в інтегральній схемі прикладної орієнтації (ASIC – Application-Specific Integrated Circuit). Для функцій пересилання кадрів і пакетів використовується одна мікросхема ASIC. Багаторівневі комутатори часто зберігають або додають у кеш дані маршрутизації по джерелу і призначенню, отримані з першого пакету в діалозі. Наступним пакетам не доводиться виконувати пошук у таблиці маршрутизації тому, що вони знаходять дані маршрутизації в пам'яті. Таким чином, кешування збільшує продуктивність цих пристроїв.

Типи комутації

Коли з'явилася комутація, комутатори підтримували два методи пересилання кадру з одного порту на інший: пересилання з буферизацією (store and forward) і комутація без буферизації (cut-through switching). Кожен з методів має свої переваги і недоліки.

Комутація з буферизацією. При використанні цього типу комутації повний кадр зчитується і зберігається в пам'яті перед передачею пристрою призначення. Комутатор перевіряє цілісність бітів в кадрі, обчислюючи значення циклічного контролю парності (CRC – Cyclic Redundancy Check). Якщо розраховане значення CRC збігається зі значенням у полі CRC кадру, комутатор пересилає кадр через порт призначення. Комутатор не пересилає кадри, якщо значення CRC не збігаються. Значення CRC знаходиться в полі контрольної послідовності кадру (FCS – Frame Check Sequence) в кадрі Ethernet.

Хоча цей метод дозволяє запобігти передачі пошкоджених кадрів в інші сегменти, він викликає значні затримки. Тому комутація з буферизацією в основному використовується в середовищах з високою імовірністю виникнення помилок, наприклад у середовищах, що часто піддаються впливу електромагнітних імпульсів.

Наскрізна комутація. Інший основний метод комутації – наскрізна комутація. Наскрізна комутація включає два методи: швидке пересилання і комутація з виключенням

фрагментів. При використанні обох методів комутатор пересилає кадр, не чекаючи його повного прийому. Оскільки комутатор не обчислює і не перевіряє значення CRC, можлива передача пошкоджених кадрів.

Швидке пересилання – найшвидший метод комутації. Комутатор пересилає кадри з порту призначення відразу після зчитування MAC-адреси. Цей метод характеризується найменшим запізненням, але може пересилати пошкоджені фрагменти. Цей метод комутації найкраще працює в стабільній мережі з невеликою кількістю помилок.

При комутації з виключенням фрагментів комутатор зчитує перші 64 байти кадру перед початком пересилання цього кадру з порту призначення. Мінімальний допустимий кадр Ethernet складає 64 байти. Кадри меншого розміру, як правило, є результатом колізій і називаються кадрами з недопустимо малою довжиною. Перевірка перших 64 байт гарантує, що комутатор не перенаправляє колізійні фрагменти.

Комутація з буферизацією має найбільшу затримку, швидке пересилання – найменшу. Затримки комутації з виключенням фрагментів лежать посередині між цими методами. Комутація з виключенням фрагментів є оптимальним методом у середовищах, в яких виникає багато колізій. У якісно спроектованій мережі колізії не є проблемою, тому кращим методом є швидка комутація.

В даний момент більшість комутаторів Cisco для локальних мереж використовують метод з буферизацією. Це пов'язано з тим, що нові технології і низький час обробки дозволяють комутаторам зберігати й обробляти кадри майже так швидко, як при наскрізній комутації, але без помилок. Крім того, багато функцій вищого класу, такі як багаторівнева комутація, використовують метод комутації з буферизацією.

Крім того, деякі нові комутатори 2-го і 3-го рівнів можуть змінювати метод комутації відповідно до зміни стану мережі. Ці комутатори виконують швидке пересилання кадрів, щоб забезпечити мінімальну затримку. Незважаючи на те, що комутатор не виявляє помилки перед пересиланням кадру, помилки розпізнаються, і їхня кількість зберігається в пам'яті. Число виявлених помилок порівнюється з попередньо заданим граничним значенням.

Якщо кількість помилок перевищує граничне значення, значить комутатор передав недопустиме число помилкових кадрів. У цьому випадку комутатор переключається на метод з буферизацією. Якщо кількість помилок опускається нижче граничного значення, комутатор повертається в режим швидкого пересилання. Цей режим називається адаптивною наскрізною комутацією.

Безпека комутаторів

Незалежно від методу комутації потрібно підтримувати безпеку мережі. Засоби мережевої безпеки часто зосереджуються на маршрутизаторах і блокуванні трафіку ззовні. Комутатори є внутрішніми пристроями організації і розроблені для зручного доступу, тому до них застосовуються тільки найпростіші заходи безпеки, або не застосовуються взагалі.

Потрібно використовувати наступні базові заходи безпеки комутатора, щоб доступ до нього могли отримати тільки авторизовані співробітники:

– захист фізичного доступу до пристрою;
– використання безпечних паролів;
– використовувати доступ через SSH;
– відслідковувати доступ і трафік;
– відключити доступ через http;
– відключити порти, що не використовуються;
– включити захист портів;
– відключити Telnet.

Використання Cisco IOS CLI

В CLI Cisco IOS [11] є безліч функцій, які допомагають викликати необхідні команди конфігурації. Наявність таких функцій пояснює, чому фахівці з обслуговування мереж воліють використовувати Cisco IOS CLI для налаштування комутаторів. При налаштуванні пристроїв особливо корисною є функція виклику контекстної довідки. Якщо ввести в командний рядок help або ?, то з'явиться короткий опис довідкової системи.

Switch# help

Контекстно-залежна довідка дозволяє отримати пропозиції по виконанню тих чи інших команд. Якщо відома не вся команда, а тільки кілька її перших символів, потрібно їх ввести, а після відомих символів поставити знак «?». Між символами команди та знаком «?» не повинно бути пробілу.

Крім того, щоб отримати список можливих параметрів для певної команди, потрібно ввести частину цієї команди, потім пробіл, а після нього знак «?». Наприклад, якщо ввести команду configure, потім пробіл і знак «?», то буде виведений список можливих варіантів продовження цієї команди. Щоб закінчити рядок команди, потрібно вибрати один з варіантів. Після завершення командного рядка з'явиться символ <cr>. Для введення команди потрібно натиснути клавішу Enter. Якщо знаку «?» не відповідає нічого із вмісту довідки, список буде порожнім. Це означає, що введена команда не підтримується.

Іноді користувачі вводять команди з помилкою. Якщо команда введена не повністю або її не вдається розпізнати, з'явиться відповідне повідомлення CLI. Символом «%» позначається початок повідомлення про помилку. Наприклад, якщо введено команду interface без додаткових параметрів, то з'явиться повідомлення про помилку, яке вказує, що команда введена не повністю:

% Incomplete command («Команда не завершена»)

Для отримання списку доступних параметрів потрібно використовувати символ «?».

Якщо команда введена неправильно, з'явиться таке повідомлення:

% Invalid input detected («Недопустима команда»)

Іноді важко помітити помилку в неправильно введеній команді, але в CLI є індикатор помилок. У тому місці стрічки команди, де перебуває неправильний або нерозпізнаний символ, з'являється знак вставки «^». Завдяки цьому, користувач може повернутися до потрібного місця та визначити правильну команду за допомогою функції довідки. Крім цього, в Cisco IOS CLI є функція виклику раніше введених команд. Ця функція особливо зручна при введенні довгих або складних команд.

Збереження історії введення команд включається за замовчуванням і система фіксує 10 записів командних рядків у буфері. Щоб змінити кількість командних рядків, які записуються системою протягом сеансу, можна використати команду terminal history size або history size. Максимальна кількість командних рядків - 256.

Для виклику з буфера останньої введеної команди можна використати клавіші Ctrl-P або клавішу зі стрілкою "вверх". Для виклику наступних команд потрібно повторити процедуру. Щоб повернутися до новішої команди з буферу, можна використати клавіші Ctrl-N або клавішу зі стрілкою "вниз". Для виклику наступних команд потрібно повторити процедуру.

CLI розпізнає частково введені команди, знаходячи перший унікальний символ. Наприклад, можна ввести «int» замість «interface». Якщо введено скорочену назву, наприклад «int», то при натисканні клавіші Tab запис команди буде автоматично доповнено до «interface».

В більшості комп'ютерів є додаткові функції вибору та копіювання за допомогою різних функціональних клавіш. Попередній рядок команди можна скопіювати та вставити як поточну команду.

Використання команд Show

Cisco IOS CLI дозволяє користуватися командами показу для відображення інформації про конфігурацію та режим роботи пристрою.

Адміністратори мережі широко користуються командами показу для перегляду файлів конфігурації, перевірки стану інтерфейсів пристроїв та поточних процесів, а також для контролю робочого стану пристроїв. Командами show можна користуватися незалежно від способу конфігурації пристрою - CLI або SDM.

За допомогою команди show можна відобразити стан практично будь-якого процесу або функції комутатора. Найбільш відомі команди show:

show running-config

show interfaces

show arp

show ip route
show protocols
show version

Базова конфігурація

У вихідну конфігурацію пристрою Cisco IOS входить призначення імені пристрою та паролів, які служать для контролю доступу до різних функцій пристрою.

Одним з перших завдань конфігурування є присвоєння пристрою унікального імені. Це завдання вирішується в режимі глобальної конфігурації за допомогою такої команди:

Switch(config)# hostname <ім'я>

При натисканні клавіші Enter ім'я вузла за замовчуванням - Switch - змінюється на нове, привласнене вузлу, ім'я.

Наступним кроком конфігурування є призначення паролів для запобігання несанкціонованого доступу до пристрою.

Для обмеження доступу до привілейованого режиму EXEC служать команди enable password та enable secret. Це не дає можливості змінювати параметри налаштування комутатора користувачам, які не мають відповідних прав доступу.

Switch (config)# enable password <пароль>

Switch (config)# enable secret <пароль>

Різниця між цими двома командами полягає в тому, що команда enable password за замовчуванням не є зашифрованою. Якщо після команди enable password вводиться команда enable secret <пароль>, то команда enable secret має перевагу перед enable password. Не зашифрований пароль можна відновити в режимі відновлення паролю.

До інших основних налаштувань комутатора належать налаштування баннера, включення синхронного ведення журналу та відключення пошуку домена.

Банери

Банер – це текст, який бачить користувач при вході в комутатор. Налаштування відповідного банера є частиною продуманого плану забезпечення безпеки. Банер повинен містити попередження щодо несанкціонованого доступу. Ніколи не встановлюйте банер у вигляді вітання для користувача, що не має відповідних прав доступу.

Існує два типи банерів: повідомлення дня (MOTD) та інформація для входу. Два окремих банери потрібні для того, щоб можна було замінити один з них, не зачіпаючи при цьому повідомлення банера цілком.

Для налаштування банерів служать команди banner motd й banner login. Для обох типів як роздільник на початку та наприкінці повідомлення використовується символ «#». Цей символ дозволяє користувачеві задати банер, що складається з декількох рядків.

Якщо задані обидва банери, то банер входу в систему з'являється після MOTD, але перед введенням облікових даних для входу.

Синхронне ведення журналу

Програма Cisco IOS часто надсилає повідомлення, наприклад, про зміну стану інтерфейсу, що конфігурується. Іноді це відбувається під час введення команди. Таке повідомлення не впливає на виконання команди, але дезорієнтує користувача, що вводить команду. Для того, щоб під час введення команди не з'являлися повідомлення, можна ввести команду logging synchronous в режимі глобальної конфігурації.

Відключення пошуку домена

За замовчуванням при введенні імені вузла в режимі включення комутатор інтерпретує це як спробу користувача підключитися до пристрою через Telnet. Комутатор намагається розв'язати невідомі імена, введені в режимі включення, шляхом відправлення їх на сервер DNS. Це відноситься до всіх введених слів, які комутатор не може розпізнати, включаючи неправильно введені команди. Якщо цього робити не потрібно, то за допомогою команди no ip domain-lookup можна відключити цю функцію, яка працює за замовчуванням.

Існує кілька способів підключитися до пристрою та налаштувати конфігурацію. Один з них - підключення комп'ютера до консольного порту пристрою. Цей тип підключення часто використовується для налаштування початкової конфігурації пристрою.

Встановлення паролю для доступу до консолі виконується в режимі глобальної конфігурації. Зазначені нижче команди запобігають несанкціонованому доступу до користувацького режиму з порту консолі.

Switch (config)# line console 0

Switch (config)# password <пароль>

Switch (config)# login

Якщо пристрій з'єднаний з мережею, то до нього можна отримати доступ через мережеве з'єднання. Такий варіант називається підключенням через віртуальний термінал або підключенням vty. Для віртуального порту (порту vty) потрібно призначити пароль.

Switch (config)# line vty 0 4

Switch (config)# password <пароль>

Switch (config)# login

Цифри 0 4 означають 5 одночасних внутрішньосмугових підключень. Можна для кожного підключення задати свій пароль, вказавши номери конкретних ліній, наприклад, line vty 0.

Для перевірки правильності призначення паролів можна використати команду show running-config. Паролі зберігаються у файлі поточної конфігурації, у форматі незашифрованого тексту. Можна включити шифрування всіх паролів, які зберігаються в пам'яті комутатора. Це створить додаткові труднощі у випадку несанкціонованого доступу.

Команда service password encryption, введена в режимі глобальної конфігурації, забезпечує шифрування всіх паролів.

Варто пам'ятати, що при зміні поточної конфігурації необхідно скопіювати її у файл початкової конфігурації. В протилежному випадку при вимиканні пристрою всі зміни будуть втрачені. Для копіювання поточної конфігурації у файл початкової конфігурації використовується команда: copy run start.

Конфігурація комутатора

Включення комутатора

Для включення комутатора необхідно виконати три основних кроки.

Крок 1. Перевірка компонентів.

Крок 2. Підключення кабелів до комутатора.

Крок 3. Запуск комутатора.

Після запуску комутатора починається його самотестування при включенні живлення (POST). У ході POST проводиться серія перевірок функцій комутатора. Індикатори мерехтять.

POST закінчується, коли світлоіндикатор SYST починає швидко мерехтіти зеленими кольорами. Якщо в процесі POST відбувається збій, індикатор SYST стає жовтим. Якщо комутатор не може виконати POST, необхідно відправити його для ремонту. Після завершення всіх стартових процедур можна приступати до налаштування комутатора Cisco 2960.

Початкова конфігурація комутатора

Існує кілька способів налаштування і керування комутатором Cisco у локальних мережах:

– Cisco Network Assistant (Мережевий помічник Cisco);

– Device Manager (SDM) (Диспетчер пристроїв Cisco);

– інтерфейс командного рядка Cisco IOS;

– програма керування CiscoView;

– програмні продукти керування мережею SNMP.

У деяких із цих способів для підключення до комутатора використовується IP-адресація або веб-браузер, що вимагає наявності IP-адреси. На відміну від інтерфейсів комутатора, портам комутатора не присвоюють IP-адресу. Щоб скористатися засобом керування на основі IP або відкрити сеанс Telnet для роботи з комутатором Cisco, потрібно налаштувати для керування IP-адресу комутатора.

Якщо в комутатора нема IP-адреси, варто підключитися безпосередньо до порту консолі та використати для налаштування емулятор терміналу.

Налаштування комутатора Cisco Catalyst 2960 виконується на заводі-виробнику. Перед підключенням до мережі необхідно задати тільки основну інформацію про безпеку.

Команди, що служать для налаштування на комутаторі імені вузла та паролів, є тими ж командами, які використовуються для налаштування ISR. Щоб працювати з комутатором Cisco через засоби керування на основі IP або Telnet, потрібно налаштувати для керування IP-адресу.

Для того, щоб призначити комутатору адресу, ця адреса має бути призначена інтерфейсу віртуальної локальної мережі VLAN. У мережі VLAN кілька фізичних портів можуть бути об'єднані логічно. За замовчуванням існує тільки одна мережа VLAN, що заздалегідь налаштована в комутаторі - VLAN1, і вона забезпечує доступ до функцій керування.

Щоб створити IP-адресу інтерфейсу керування VLAN1, потрібно перейти в режим глобальної конфігурації [11].

Switch>enable

Switch#configure terminal

Далі, увійти в режим конфігурації інтерфейсу для VLAN1.

Switch(config)#interface vlan 1

Задати IP-адресу, маску мережі та шлюз за замовчуванням для інтерфейсу керування. IP-адреса повинна перебувати в тій же локальній мережі, що й комутатор.

Switch(config-if)#ip address 192.168.1.2 255.255.255.0

Switch(config-if)#exit

Switch(config)#ip default-gateway 192.168.1.1

Switch(config)#end

Після цього зберегти конфігурацію за допомогою команди copy running-configuration startup-configuration.

Підключення комутатора до LAN

Підключення комутатора до мережі

Для підключення комутатора до маршрутизатора використовується прямий кабель. Світлоіндикатори на комутаторі та маршрутизаторі свідчать про успішність підключення.

Після з'єднання комутатора та маршрутизататора потрібно перевірити, чи можуть ці два пристрої обмінюватися повідомленнями.

Насамперед, потрібно перевірити налаштування IP-адреси. За допомогою команди show running-configuration потрібно переконатися в тому, що IP-адреса інтерфейсу управління комутатора мережі VLAN1 та IP-адреса безпосередньо під'єднаного інтерфейсу комутатора перебувають в одній локальній мережі.

Потім потрібно перевірити з'єднання за допомогою команди ping. Для цього з комутатора відправляється команда ping на IP-адресу безпосередньо під'єднаного інтерфейсу комутатора. Після цього потрібно повторити цей процес із комутатора, відправивши команду ping на IP-адресу інтерфейсу керування, призначену комутатору мережі VLAN 1.

Якщо ехо-запит виконати не вдалося, потрібно перевірити з'єднання та конфігурацію ще раз і переконатися в тому, що всі кабелі підключені правильно та надійно.

Коли між комутатором і маршрутизатором встановлений нормальний обмін даними, можна підключати до комутатора, за допомогою прямих кабелів, окремі ПК. Ці кабелі можуть прямо підключатися до ПК або вони можуть бути частиною структурованої кабельної системи, що йде до настінних розеток.

Порти комутатора можуть бути місцями несанкціонованого входу в мережу. Для запобігання цього комутатори підтримують функцію, що називається захистом портів. Ця функція обмежує кількість допустимих MAC-адрес на один порт. Порт не буде відправляти пакети з вихідними MAC-адресами, якщо вони не входять у групу заданих адрес. Існує три способи налаштування безпеки порту:

Статичний

MAC-адреси призначаються вручну, використовуючи команду налаштування інтерфейсу: switchport port-security mac-address <mac-адреса>

Статичні MAC-адреси зберігаються в таблиці адрес і додаються в поточну конфігурацію.

Динамічний

Динамічно отримані відомості про MAC-адреси зберігаються в таблиці адрес. Кількість отриманих адрес можна контролювати. За замовчуванням на один порт може бути отримано не більше однієї MAC-адреси. Отримані адреси видаляються з таблиці при вимиканні порту або при перезапуску комутатора.

Зв'язаний

Аналогічний динамічному, за винятком того, що адреси зберігаються ще й у поточну конфігурацію.

За замовчуванням безпека порту відключена. Якщо включити функцію безпеки порту, це приведе до несправності при відключенні порту. Наприклад, якщо включити функцію безпеки порту в динамічному режимі і на один порт може бути отримано не більше однієї MAC-адреси, то перша отримана адреса стає безпечною. Якщо інша робоча станція спробує одержати доступ до порту з іншою MAC-адресою, то відбудеться порушення безпеки і відповідний порт буде заблоковано. При цьому може надсилатись повідомлення про блокування порту. Порушення безпеки відбувається в кожній із зазначених нижче ситуацій:

– у таблицю адрес введена максимально можлива кількість безпечних MAC-адрес, і пристрій, адреси якого немає в таблиці адрес, намагається отримати доступ до інтерфейсу.

– адресу, отриману або налаштовану на одному безпечному інтерфейсі, можна бачити на іншому безпечному інтерфейсі в тій же мережі VLAN.

Щоб можна було активувати функцію безпеки порту, необхідно перевести порт у режим доступу за допомогою команди switchport mode access.

Для перевірки налаштувань безпеки порту для комутатора або заданого інтерфейсу, використовується команда show port-security interface interface-id. На екрані з'являться такі вихідні дані:

– максимально допустима кількість безпечних MAC-адрес для кожного інтерфейсу;

– кількість безпечних MAC-адрес даного інтерфейсу;

– кількість порушень безпеки, що відбулися;

– режим порушення безпеки.

Крім цього, при введенні команди show port-security address відображаються безпечні MAC-адреси для всіх портів, а при введенні команди show port-security відображаються налаштування безпеки порту для комутатора.

Якщо включено функцію безпеки порту для статичного або зв'язаного режиму, то можна використати команду show running-config для перегляду MAC-адрес, пов'язаних з конкретним портом. Існує три способи видалення отриманої MAC-адреси, що була збережена у поточну конфігурацію:

– використати команду clear port-security sticky interface <№ порту> access для видалення всіх отриманих адрес. Потім варто виключити порт, ввівши команду shutdown. Нарешті, потрібно знову включити порт за допомогою команди no shutdown.

– для відключення режиму безпеки порту варто ввести з інтерфейсу команду no switchport port-security. Після відключення варто знову включити режим безпеки порту

– перезавантажити комутатор.

Комутатор буде перезавантажуватися тільки в тому випадку, якщо поточна конфігурація не збережена у файл початкової конфігурації. Якщо ж поточна конфігурація була збережена у файл початкової конфігурації, то це виключає для комутатора можливість повторного отримання адрес при перезавантаженні системи.

Однак отримані MAC-адреси будуть завжди пов'язані з конкретним портом доти, поки не буде зроблене очищення порту за допомогою команди clear port-security, або не буде відключений режим безпеки порту. Якщо це буде зроблено, потрібно перезаписати поточну конфігурацію у файл початкової конфігурації, щоб комутатор після перезавантаження не почав використовувати вихідні зв'язані MAC-адреси.

Якщо на комутаторі є порти, що не використовуються, рекомендується відключити їх з міркувань безпеки для обмеження можливості несанкціонованого доступу. Відключення

портів на комутаторі виконується просто. Переходячи до кожного порту, що не використовується, потрібно ввести команду shutdown. Якщо потім треба буде активувати цей порт, використовується команда no shutdown для відповідного інтерфейсу.

Крім включення режиму безпеки порту та відключення портів, які не використовуються, існують інші налаштування безпеки комутатора, які дозволяють встановлювати паролі на порти vty, застосовувати банери входу в систему та зашифровувати паролі за допомогою команди service password-encryption. Для зазначених конфігурацій використовуються ті ж команди інтерфейсу командного рядка Cisco IOS, які застосовуються для налаштування комутатора.

Мінімальний розмір кадрів Ethernet складає 64 байти, максимальний – 1518 байтів, однак розмір тегованого кадру Ethernet може досягати 1522 байти.

Кадри включають такі поля:

– MAC-адреси джерела і призначення;

– довжина кадру;

– корисні дані;

– контрольна послідовність кадру (FCS).

Поле FCS забезпечує виявлення помилок і гарантує цілісність всіх бітів в кадрі.

Мітка збільшує мінімальний розмір кадру Ethernet з 64 до 68 байт. Максимальний розмір збільшується з 1518 до 1522 байт. Комутатор перерозраховує FCS, тому що кількість бітів в кадрі збільшується.

Якщо порт, сумісний з 802.1q, підключений до іншого порту, також сумісного з 802.1q, дані маркування VLAN передаються між ними.

Якщо підключений порт несумісний з 802.1q, мітка VLAN буде видалена перш, ніж кадр досягне середовища передачі.

Якщо пристрій, або порт доступу без підтримки 802.1q отримує кадр 802.1q, то дані мітки ігноруються, а пакет комутується на рівні 2 як стандартний кадр Ethernet. Це дозволяє розміщувати на транковому маршруті 802.1q проміжні пристрої рівня 2, наприклад інші комутатори. Щоб обробити кадр із міткою 802.1q, пристрій повинен дозволяти MTU зі значенням 1522 або вище.

Призначення протоколу VTP

Автоматизація керування VLAN

Зі збільшенням розміру і складності мережі централізоване керування структурою VLAN стає критично важливим. Протокол VTP (VLAN Trunking Protocol) – це протокол обміну повідомленнями 2-го рівня, що надає метод керування базою даних VLAN з центрального сервера в мережевому сегменті. Маршрутизатори не пересилають оновлення VTP.

Без автоматизованого методу керування корпоративною мережею із сотнями VLAN потрібно було б вручну налаштувати кожну VLAN на кожному комутаторі. Будь-яка зміна структури VLAN потребувала б додаткового ручного налаштування. Один невірно набраний номер може стати причиною нестабільності з’єднань по всій мережі.

Щоб вирішити цю проблему, корпорація Cisco створила протокол VTP, що автоматизує багато задач конфігурації VLAN. VTP гарантує погоджене обслуговування конфігурації VLAN по всій мережі і зменшує необхідність у керуванні та моніторингу VLAN.

VTP являє собою протокол обміну повідомленнями між клієнтом і сервером, що дозволяє додавати, видаляти та перейменовувати VLAN в одному домені VTP. Усі комутатори під загальним керуванням є частиною домена. У кожного домена є унікальне ім’я. Комутатори VTP обмінюються повідомленнями VTP тільки з іншими комутаторами в домені.

Компоненти протоколу VTP

Ключовими компонентами протоколу VTP є:

VTP домен – складається з одного або декількох взаємопов’язаних комутаторів. Всі комутатори в домені діляться інформацією про деталі налаштування VLAN за допомогою

VTP оголошень. Маршрутизатор або комутатор третього рівня визначає межі кожного домену.

VTP оголошення – VTP використовує оголошення для поширення та синхронізації конфігурації VLAN по всій мережі.

Режими VTP – комутатор може бути налаштований на роботу в одному з трьох режимів: сервер, клієнт, або прозорий.

VTP сервер – VTP сервер поширює інформацію про VLAN іншим комуторам, що підтримують VTP, в тому ж VTP домені. VTP сервери зберігають інформацію про VLAN для всього домену в енергонезалежній пам'яті. На сервері VLAN можуть бути створені, видалені або перейменовані.

VTP клієнт – VTP клієнти працюють так само, як VTP сервери, але на них не можна створювати, змінювати, або видаляти VLAN. VTP клієнт зберігає інформацію про VLAN для всього домену лише в той час коли комутатор включений. Перезавантаження комутатора спричиняє видалення інформації про VLAN. Режим VTP клієнта на комутаторі необхідно налаштувати.

VTP прозорий – прозорий комутатор пересилає VTP повідомлення до VTP клієнтів та серверів. Прозорі комутатори не приймають участь в роботі VTP. Мережі VLAN, які створюються, перейменовуються або видаляються на прозорих комутаторах є локальними і відносяться лише до цього комутатора.

VTP Pruning(скорочення) – збільшує доступну смугу пропускання мережі шляхом обмеження поширення інформації про VLAN комутаторові у якого немає активних портів у відповідному VLAN, а також блокує широкомовний трафік на комутатор, якщо в ньому немає активних портів у тому ж VLAN з якого відправляється широкомовний трафік.

Структура VTP кадру

VTP оголошення (або повідомлення) поширюють ім'я домену VTP та зміни в конфігурації VLAN до комутаторів, які підтримують VTP.

VTP кадр складається з заголовка та повідомлення. VTP інформація вставляється в поле даних кадру Ethernet. Після цього Ethernet кадр інкапсулюється в транковий кадр 802.1q (або ISL кадр). Кожен комутатор в домені періодично надсилає повідомлення на кожен транковий на спеціальну групову адресу. Ці оголошення приймаються сусідніми комутаторами, які оновлюють свої VTP та VLAN конфігурації в міру необхідності.

Інкапсуляція VTP кадру в 802.1q кадр не є статичною. Зміст VTP повідомлення визначається наявними полями. Комутатори з підтримкою VTP шукають визначені поля та значення в кадрі 802.1q. В інкапсульованому у 802.1.q кадр VTP кадрі присутні наступні ключові поля:

Destination MAC address – MAC-адреса призначення, встановлена в значення 01-00-0C-CC-CC-CC, яке є зарезервованим значенням групової адреси для всіх VTP повідомленнь.

LLC (Logical link control) field – поле містить сервісну точку доступу призначення (destination service access point, DSAP) та сервісну точку доступу відправника (source service access point, SSAP) встановлені в значення AA.

SNAP (Subnetwork Access Protocol) field – поле з встановленим OUI в значення AAAA і встановленим типом 2003.

VTP header field – поле VTP заголовку, вміст змінюється в залежності від типу VTP повідомлення, але завжди містить такі поля VTP поля:

– Domain name – визначає адміністративний домен для комутатора;

– Domain name length – довжина доменного імені;

– Version – в-я, яку підтримує комутатор, встановлюється VTP 1, VTP 2, або VTP 3;

– Configuration revision number –поточний номер версії конфігурації на комутаторі.

VTP message field – значення залежить від типу повідомлення.

VTP Message Contents – VTP кадри містять наступну глобальну інформацію про домен фіксованої довжини:

– VTP domain name;

– ідентифікатор комутатора, що надіслав повідомлення та час його надсилання;

– MD5 значення VLAN конфігурації, включаючи максимальний блок передачі

(maximum transmission unit, MTU) для кожноїVLAN;

– Формат кадру: ISL або 802.1q.

VTP кадри містять наступну інформацію для кожної налаштованої VLAN:

– VLAN IDs (IEEE 802.1q);

– VLAN name;

– VLAN type;

– VLAN state;

– додаткова інформація про налаштування VLAN.

Режими роботи VTP

Існує дві різні версії VTP: 1 і 2. Версія 1 – версія за замовчуванням і вона несумісна з версією 2. На всіх комутаторах необхідно налаштувати однакову версію протоколу.

VTP використовує три режими: сервер, клієнт і прозорий пристрій. За замовчуванням усі комутатори є серверами. Рекомендується налаштувати хоча б два комутатори в мережі як сервери, щоб забезпечити резервування. Ці режими відрізняються тим, як вони використовуються для управління та надсилання повідомлень про VTPдомени та VLAN.

Режим сервер

В режимі сервера можна створювати, змінювати та видаляти VLAN для всього домену VTP. Режим VTP сервера використовується за замовчуванням на комутаторах Cisco. VTP сервери надсилають свої конфігурації VLAN на інші комутатори в тому ж домені VTP та синхронізують конфігурації VLAN з іншими комутаторами по магістральних каналах. VTP сервери відстежують оновлення через номер версії конфігурації. Інші комутатори в тому ж домені VTP порівнюють номер версії своєї конфігурації з номером ревізії, який було отримано з VTP сервера, щоб визначити чи вони повинні синхронізувати свою базу даних VLAN.

Режим клієнт

Якщо комутатор знаходиться в режимі клієнта, на ньому не можна створити, змінити або видалити VLAN. Крім того, інформацію про VLAN конфігурації VTP клієнт отримує від VTP сервера і вона зберігається в базі даних VLAN, а не в NVRAM. Завдяки цьому, VTP клієнти вимагають менше пам’яті, ніж VTP сервери. Після включення чи перезавантаження VTP клієнт надсилає запит на VTP сервер для оновлення інформації про конфігурацію VLAN. Комутатори налаштовані як клієнти зазвичай зустрічаються у великих мережах.

Прозорий режим

Комутатори налаштовані на прозорий режим роботи VTP пересилають VTP повідомлення, які вони отримують на магістральні порти на інші комутатори в мережі. Також вони не надсилають свою конфігурацію VLAN і не синхронізують конфігурацію VLAN з будь-яким іншим комутатором.

У прозорому режимі VLAN конфігурації зберігаються венергонезалежній пам’яті, тому конфігурація доступна після перезавантаження. Також це означає, що після перезавантаженнія комутатор не повернеться до режим VTP сервера pf замовчуванням, а залишається в прозорому режим VTP.

При використанні VTP кожен сервер посилає повідомлення через свої транкові порти. Повідомлення включають домен керування, номер версії конфігурації, відомі VLAN і параметри кожної VLAN. Кадри оголошень відправляються за адресою багатоадресної розсилки, тому їх отримують всі сусідні вузли.

Кожен комутатор VTP зберігає базу даних VLAN, що включає номер версії конфігурації, в енергонезалежній пам’яті (NVRAM). Якщо VTP отримує оновлення з більш високим номером версії, ніж номер у базі даних, комутатор додає нові дані у свою базу даних VLAN.

Номер зміни конфігурації VTP починається з нуля. При внесенні змін номер версії конфігурації збільшується на одиницю. Номер версії продовжує збільшуватися, поки не досягає 2 147 483 648. При досягненні цього значення лічильник скидається на нуль. Крім того, номер версії скидається при перезавантаженні комутатора.

Проблема, пов’язана з номером версії, може виникнути, якщо додати у мережу комутатор з більш високим номером версії. Оскільки за замовчуванням комутатор

знаходиться в серверному режимі, нові, але невірні дані можуть перезаписати коректні дані VLAN на всіх інших комутаторах.

Один зі способів забезпечення захисту від цієї критичної ситуації полягає в заданні паролів VTP для перевірки комутаторів. Крім того, при додаванні комутатора в мережу, у якій уже є комутатор у серверному режимі, переконайтеся, що новий комутатор налаштований у прозорому або клієнтському режимі.

Повідомлення та налаштування VTP

Існує три типи повідомлень VTP: зведені оголошення, скорочені оголошення і запити оголошень.

Зведені оголошення

Комутатори Catalyst розсилають зведені оголошення кожні 5 хвилин, а також при зміні бази даних VLAN. Зведені оголошення містять поточне ім'я домена VTP і номер версії конфігурації.

При додаванні, видаленні або зміні VLAN сервер збільшує номер версії конфігурації і відправляє зведене оголошення.

При отриманні пакета зведеного оголошення комутатор порівнює ім'я домена VTP зі своїм ім'ям домену VTP. Якщо імена доменів збігаються, комутатор порівнює номер версії конфігурації зі своїм номером. Якщо отриманий номер менший, комутатор ігнорує пакет. Якщо номер версії вищий, відправляється запит оголошення.

Скорочені оголошення

Скорочене оголошення відправляється після зведеного оголошення. Скорочене оголошення містить список даних VLAN.

Скорочене оголошення містить нові дані VLAN, засновані на зведеному оголошенні. Якщо в мережі кілька VLAN, буде потрібно кілька скорочених оголошень. Причиною оновлення скороченого повідомлення може бути:

– створення або видалення VLAN;

– призупинення або активація VLAN;

– зміна імені VLAN;

– зміна MTU VLAN.

Для повного оновлення інформації про VLAN потрібно використати декілька скорочених повідомлень.

Оголошення-запити.

VTP-клієнти використовують оголошення-запити, щоб запитати інформацію про VLAN. Оголошення-запити необхідні, якщо комутатор скинутий або змінене ім'я домена VTP. Комутатор отримує зведене оголошення VTP з більш високим номером версії конфігурації, ніж його власний.

Коли оголошення-запит надходить VTP серверу в тому ж VTP домені, VTP сервер відповідає, надсилаючи зведене оголошення, а потім скорочене оголошення.

Оголошення-запити направляються в таких випадках:

– після зміни імені VTP домену;

– при отримані комутатором зведеного оголошення з більш високим номером версії конфігурації ніж його власний;

– скорочене повідомлення втрачене з якихось причин;

– комутатор ресетувався.

За замовчуванням комутатори є VTP серверами. Якщо комутатор у серверному режимі відправляє відновлення з номером версії, що перевищує поточний номер версії, усі комутатори змінять свої бази даних відповідно до отриманого повідомлення.

При додаванні нового комутатора в існуючий домен VTP потрібно виконати наступні дії:

– дія 1. Налаштувати протокол VTP в автономному режимі (версію 1);

– дія 2. Перевірити конфігурацію VTP;

– дія 3. Перезавантажити комутатор.

При налаштуванні VTP можуть виникати певні проблеми.

Несумісність версій VTP – версії 1 та версії 2 несумісні одна з одною, тому старі комутатори, які підтримують тільки VTP версії 1 не можуть працювати у VTP домені поряд з комутаторами з версією 2. Якщо в мережі є комутатори, які підтримують тільки версію 1 потрібно вручну налаштувати комутатори з версія 2 для роботи в режимі версії 1

Неспівпадіння VTP паролів – при використанні VTP паролю для контролю участі в домені VTP, пароль повинен бути встановлений правильно на всіх комутаторах в домені VTP. На відміну від інших параметрів, які встановлюються автоматично піля отримання VTP повідомлення, комутатор не виконує автоматичне налаштування параметрів паролів.

Невірне ім'я VTP домену – ім'я домену VTP є ключовим параметром, який встановлюється на комутаторі. Неправильно налаштований VTP домен впливає на синхронізацію VLAN між комутаторами. Якщо комутатор отримує неправильне VTP оголошення, він його. Якщо відкинуте повідомлення містить інформацію про конфігурацію, комутатор не синхронізує свою базу данихVLAN, як очікувалося.

Щоб уникнути неправильного налаштування VTP домен, потрібно встановити ім'я VTP домену на комутаторі, який є VTP сервером. Всі інші комутатори в тому ж домені VTP будуть приймати та автоматично налаштувати їх VTP доменне ім'я, коли отримають перше зведене VTP оголошення. Всі інші комутатори можна встановити в режим VTP клієнтів. При цьому втрачається можливість для створення, видалення та управління VLAN в мережі. Оскільки комутатори в режимі VTP клієнта не зберігають інформацію про VLAN в енергонезалежній пам'яті, вони повинні оновити інформацію про VLAN після перезавантаження.

Щоб не втратити всі VLAN налаштування в VTP домені внаслідок випадкового переналаштування VTP сервера в режим VTP клієнта, можна налаштувати ще один комутатор в тому ж домені в режим сервера VTP.

Запобігання утворенню петель комутації

Резервування в мережі

Сучасні підприємства усе більше покладаються на мережі, іноді від мереж залежить саме їх існування. Мережа – життєво важлива комунікація для багатьох організацій. Простій мережі перетворюється у катастрофічні втрати для бізнесу і довіри замовників.

Відмова одного мережевого каналу, одного пристрою чи навіть важливого порта комутатора може стати причиною простою мережі. Щоб виключити критичні точки відмови і забезпечити високу надійність, у мережеву архітектуру необхідно ввести резервування. Резервування реалізується шляхом встановлення дубльованого обладнання та мережевих пристроїв на важливих ділянках.

Іноді повне резервування всіх каналів і пристроїв стає невиправдано дорогим. Мережеві інженери часто змушені шукати компроміс між витратами на резервування і вимогами до доступності мережі.

Резервування означає наявність двох різних шляхів до одного місця призначення. Якщо один шлях заблокований, другий залишається доступним.

Резервування комутаторів реалізується шляхом створення декількох каналів між ними. Резервні канали в мережі знижують перевантаження і підтримують високу доступність і розподіл навантаження.

Вплив режимів передачі трафіку

Однак з'єднання комутаторів може стати причиною проблем. Зокрема, широкомовна природа трафіку Ethernet приводить до утворення петель комутації. Широкомовні кадри циклічно поширюються у всіх напрямках, викликаючи «шторм» широкомовних пакетів. Широкомовні шторми займають усю доступну смугу пропускання, блокують створення нових мережевих підключень і розривають існуючі підключення.

Широкомовні шторми – не єдина проблема, що обумовлена резервними каналами в комутованій мережі. Кадри одноадресного пересилання можуть також викликати такі проблеми, як множинна передача кадрів і нестабільність бази даних MAC-адрес.

Множинна передача кадрів

Якщо вузол посилає одноадресний кадр вузлу призначення і MAC-адреса не представлена в жодній з таблиць MAC-адрес підключених комутаторів, усі комутатори

виконують лавинне розсилання цього кадру з усіх портів. У мережі з петлями кадр може повернутися до вихідного комутатора. Цей процес повторюється, що приводить до утворення декількох копій кадру в мережі. В результаті вузол призначення отримує кілька копій кадру. Це стає причиною трьох проблем: неефективна витрата смуги пропускання, неефективна витрата циклів ЦП і дублювання трафіку.

Нестабільність бази даних MAC-адрес

Комутатори в резервованій мережі можуть отримувати невірні дані про місцезнаходження вузла через наявність петель комутаціїї. Якщо в мережі присутня петля, один комутатор може зв'язати MAC-адресу призначення з двома портами в своїй таблиці MAC-адрес. Це приведе до плутанини і неоптимального пересилання кадрів та збільшення завантаження каналів передачі даних і навантаження на комутатор.

Протокол STP забезпечує механізм відключення резервних каналів в комутованій мережі. STP дозволяє використовувати резервування, необхідне для надійної експлуатації, без створення петель комутації. STP ґрунтується на відкритих стандартах і використовується для створення логічної топології без петель комутації.

Протокол STP відносно самодостатній і вимагає мінімального налаштування. При першому включенні комутатори з підтримкою STP перевіряють мережу на наявність петель. Комутатори при виявленні петлі, блокують деякі з підключених портів, залишаючи інші порти активними для пересилання кадрів.

STP задає дерево, що охоплює всі комутатори в топології "розширена зірка". Комутатори постійно перевіряють мережу, щоб гарантувати відсутність петель і ефективну роботу всіх портів.

Щоб запобігти утворенню петель, протокол STP:

– переводить частину інтерфейсів в резервний або заблокований режим;

– залишає інші інтерфейси в режимі пересилки;

– переналаштовує мережу, активуючи відповідний резервний шлях, якщо шлях пересилки стає недоступним.

У термінології STP термін "комутатор" часто замінюється терміном "міст". Наприклад, кореневий міст – це основний міст або центральна вузол в топології STP. Кореневий міст взаємодіє з іншими комутаторами за допомогою блоків даних протоколу моста (bridge protocol data unit, BPDU). BPDU – це кадри, що розсилаються іншим комутаторам кожні 2 секунди. BPDU містять наступні відомості:

– ідентифікатор комутатора-джерела;

– ідентифікатор порту-джерела;

– сукупна вартість маршруту до кореневого моста;

– значення таймерів старіння;

– значення таймера "вітання".

При включенні комутатора кожен порт проходить через послідовність з 4 режимів: блокування, прослуховування, навчання і пересилання. П'ятий режим, "відключений", вказує на те, що адміністратор відключив порт комутатора.

Порт послідовно проходить через ці режими, при цьому колір світлодіодних індикаторів змінюється від мерехтливого жовтогарячого до немерехтливого зеленого. Проходження через режими STP може зайняти до 50 секунд, після чого комутатор буде готовий до пересилання кадрів.

При включенні комутатор переходить у режим блокування, щоб запобігти негайному утворенню петель. Потім він переходить у режим прослуховування, в якому приймає BPDU від сусідніх комутаторів. Після обробки цієї інформації комутатор визначає, які порти можуть пересилати кадри, не формуючи петлі. Якщо порт може пересилати кадри, він переходить у режим навчання, а потім у режим пересилання.

Алгоритм STP

Протокол STP використовує алгоритм сполучного дерева (Spanning Tree Algorithm, STA), щоб визначити, які порти комутатора в мережі мають бути заблоковані для запобігання виникнення петель комтації. STA визначає один комутатор в якості кореневого мосту та використовує його в якості точки відліку для всіх розрахунку всіх шляхів.

Усі комутатори приймають участь в обміні кадрами BPDU щоб визначити, який комутатор має найнижчий bridgeID (BID) в мережі. Комутатор з найменшим значенням bridgeID автоматично стає кореневим мостом для розрахунків алгоритму STA.

Кожен BPDU містить BID, який ідентифікує комутатор, що надіслав BPDU. BID містить значення пріоритету, MAC-адресу комутатора-відправника та додаткові розширені ID системи. Найнижче значення BID визначається комбінацією цих трьох параметрів.

Після визначення кореневого мосту, STA розраховує найкоротший шлях до кореневого мосту. Кожен комутатор використовує STA щоб визначити, які порти заблокувати. Протягом часу визначення найкращих шляхів до кореневого мосту для всіх напрямків у широкомовному домені, весь трафік не має можливості передаватись по мережі.

Алгоритм STA визначає вартість шляху та вартість порту при визначенні шляху, який потрібно залишити розблокованим. Вартість шляху розраховується за вартістю порту, яка, в свою чергу, пов’язана зі швидкістю порту, для кожного порту комутатора для заданого шляху. Сума значень вартості портів визначає загальну вартість шляху до кореневого мосту. Якщо є більше ніж один шлях, STA вибирає шлях з найменшою вартістю шляху.

Коли STA визначив, які шляхи повинні залишатися доступними, він налаштовує різні ролі портам комутатора. Ролі портів описують зв’язок з кореневим мостом та можливість пересилання трафіку.

Всі комутатори в широкомовному домені беруть участь у виборчому процесі. Після включення комутатор надсилає BPDU кадри, що містять BID комутатора та rootID кожні 2 секунди. За замовчуванням, rootID відповідає BID для усіх комутаторів в мережі. RootID ідентифікує кореневий міст в мережі. Спочатку кожен комутатор ідентифікує себе як кореневий міст.

У кожній мережі працює тільки один кореневий міст, що вибирається на підставі ідентифікатора моста (BID, Bridge ID). BID дорівнює сумі значення пріоритету моста та його MAC-адреси.

Значення пріоритету моста за замовчуванням дорівнює 32 768. Якщо MAC-адреса комутатора AA-11-BB-22-CC-33, BID буде дорівнювати 32768:AA-11-BB-22-CC-33.

Міст із найменшим значенням BID стає кореневим. Оскільки комутатори, як правило, використовують однакове значення пріоритету за замовчуванням, комутатор з найменшою MAC-адресою стає кореневим мостом.

При включенні комутатор припускає, що є кореневим мостом, і розсилає кадри BPDU зі своїм ідентифікатором BID. Наприклад, якщо комутатор S2 повідомляє, що кореневий ідентифікатор менший, ніж ідентифікатор S1, S1 припиняє оголошення свого ідентифікатора моста і приймає кореневий ідентифікатор S2. S2 стає кореневим мостом.

STP використовує три типи портів: кореневі порти, призначені порти і заблоковані порти.

Порт із маршрутом оптимальної вартості до кореневого моста призначається кореневим. Комутатори обчислюють шлях з найменшою вартістю, використовуючи вартість смуги пропускання кожного каналу на шляху до кореневого моста.

Призначений порт пересилає трафік до кореневого моста, але не підключений до шляху з найменшою вартістю.

Заблокований порт не пересилає трафік.

Перед налаштуванням STP мережевий адміністратор планує та оцінює мережу, щоб вибрати комутатор, який буде оптимальним кореневим мостом STP. Якщо кореневий міст буде обраний за мінімальною MAC-адресою, пересилання може бути неоптимальне.

В ролі кореневого мосту найкраще буде працювати комутатор, розташований у центрі мережі. Блокування порту, розташованого на периферії мережі, приведе до того, що трафік буде передаватися до місця призначення по довшому маршруту, ніж при використанні комутатора в центрі мережі.

Щоб задати кореневий міст, для BID обраного комутатора налаштовується мінімальний пріоритет. Для налаштування пріоритету моста використовується команда bridge priority. Значення пріоритету може знаходитися в діапазоні від 0 до 65 535, але крок між значеннями складає 4 096. Значення за замовчуванням - 32 768.

Найкращий шлях до кореневого моста.

Після призначення кореневого мосту алгоритм STA починає процес визначення найкращого шляху до кореневого мосту з усіх напрямків в широкомовному домені. Оптимальний шлях визначається шляхом підсумовування вартості портів на шляху від призначення до кореневого мосту.

Вартість порту за замовчуванням визначається швидкістю, з якою працює порт. Так, наприклад, 10-Гбіт/с Ethernet порт має вартість 2, 1 Гбіт/с порт має вартість, 100 Мбіт/с порт має вартість 19 і 10 Мбіт/с Ethernet порт має вартість порту 100.

З появою нових, більш швидких технологій Ethernet, значення вартості шляху може змінитися.

Хоча порти комутатора мають вартість порту за замовчуванням, вартість порту можна змінити. Можливість налаштування вартості портів дозволяє адміністратору гнучко керувати шляхами сполучного дерева до кореневого мосту.

Для налаштування вартості порту потрібно ввести значення вартості за допомогою команди spanning-tree cost в режимі конфігурації інтерфейсу. Діапазон значень може бути від 1 до 200 000 000. Для повернення до стандартного значення використовується команда no spanning-tree cost.

STP визначає логічний маршрут без утворення петлі в межах широкомовного домену. Дерево будується на основі інформації, отриманої шляхом обміну BPDU кадрів між комутаторами. Кожен порт комутатора переходить через п’ять можливих станів порту і три BPDU таймери.

Побудова STP дерева розпочинається відразу після завершення завантаження. Якщо порт комутатора перейде від блокування безпосередньо в стан пересилання, порт може тимчасово створити петлю, якщо комутатор не володіє всією інформацією про топологію. З цієї причини в STP передбачено п’ять станів портів.

Blocking(блокування) – порт не є призначеним портом і не бере участі в передаванні кадрів. Порт отримує кадр BPDU, щоб визначити місце розташування root ID кореневого моста і який стан кожного порту комутатора повинен бути після завершення побудови активної STP топології.

Listening (прослуховування) – протокол STP встановив, що порт може приймати участь в пересиланні кадрів відповідно до попередньо отриманого кадру BPDU. Порт комутатора може не тільки отримувати кадри BPDU, але також і передавати свої власні кадри BPDU та інформувати сусідні комутатори, що порт комутатора готується до участі в активній топології.

Learning (навчання) – порт готується приймати участь в пересиланні кадрів і починається заповнення таблиці MAC-адрес.

Forwarding (пересилання) – порт вважається частиною активної топології і пересилає фрейми, а також відправляє і отримує кадри BPDU.

Disabled (відключений) – порт не приймає участі в Spanning Tree і не пересилає фрейми. Відключений стан встановлюється, коли порт комутатора відключений адміністративно.

Час, протягом якого порт залишається в різних станах, визначається BPDU таймерами. Тільки комутатор, який є кореневим мостом може відправляти інформацію по дереву для налаштування таймерів. Продуктивність STP і зміни стану визначають такі таймери:

– Hello time;

– Forward delay;

– Maximum age.

Коли STP дозволений, кожен порт комутатора проходить через заблокований стан, проміжні стани прослуховування та навчання при включенні живлення. Порти потім стабілізуються в стані пересилання або блокування. Під час зміни топології, порт тимчасово реалізовує прослуховування і навчання на певний період, так званий інтервал затримки пересилання.

Ці значення інтервалів забезпечують адекватний час для збіжності в мережі з діаметром сім комтаторів. Це кількість комутаторів, які повинен пройти кадр з двох дальніх точок широкомовного домену. Діаметр на сім комутаторів – це найбільший дозволений діаметр, який забезпечує допустимий час збіжності протоколу STP. Конвергенція по відношенню сполучного дерева є часом, що витрачається на перерахунок сполучного дерева, якщо виникають проблеми з комутатором або лінією зв'язку.

Після визначення кореневого моста, а також кореневих, призначених та заблокованих портів, STP розсилає кадри BPDU по мережі із 2-секундним інтервалом. STP продовжує відслідковувати ці BPDU, щоб переконатися у відсутності каналів, які відмовили, і нових петель.

Якщо відбувається відмова каналу, STP перерозраховується шляхом:

– переведення деяких портів із заблокованого режиму в режим пересилання;

– переведення деяких портів з режиму пересилання в режим блокування;

– формування нового дерева STP для запобігання утворення петель у мережі.

Час очікування деяких прикладних програм може бути меншим періоду перерахунку, що може привести до зниження продуктивності. Частий перерахунок STP негативно впливає на час роботи систем.

Високопродуктивний корпоративний сервер підключається до порту комутатора. Якщо для цього порту виконується перерахунок через STP, сервер буде недоступний протягом хвилини. Важко уявити, яка кількість транзакцій може бути загублена за цей час.

У стабільній мережі перерахунки STP відбуваються рідко. Якщо мережа не стабільна, необхідно перевірити стабільність комутаторів та зміни їх конфігурацій. Одна з найпоширеніших причин перерахунків STP – несправне джерело живлення чи кабель живлення комутатора. Несправність джерела живлення викликає несподіване перезавантаження пристрою.

Ряд вдосконалень STP зводить до мінімуму час простоїв, викликаних перерахунком STP. До них належать режими роботи PortFast, UplinkFast і BackboneFast.

STP PortFast негайно переводить порт доступу в режим пересилання, минаючи режими прослуховування і навчання. Застосування PortFast на портах доступу, підключених до однієї робочої станції чи сервера, дозволить їм негайно підключатися до мережі, не очікуючи конвергенції STP.

STP UplinkFast прискорює вибір нового кореневого порту при відмові комутатора чи каналу, а також при перерахунку STP. Кореневий порт негайно переходить у режим пересилання, оминаючи режими прослуховування і навчання.

Протокол RSTP (Rapid Spanning Tree Protocol)

Коли інститут IEEE розробив оригінальний протокол STP (Spanning Tree Protocol) 802.1D, період відновлення розміром 1-2 хвилини був допустимим. Сьогодні комутація рівня 3 та удосконалені протоколи маршрутизації забезпечують більш швидкі альтернативні шляхи до місця призначення. Через потребу в передачі трафіку, чуттєвого до затримок, наприклад голосу і відео, мережі повинні підтримувати швидку конвергенцію, щоб задовольняти вимоги нових технологій.

Протокол Rapid Spanning Tree Protocol (RSTP), визначений у стандарті IEEE 802.1w, значно прискорює перерахунок STP. На відміну від функцій PortFast, UplinkFast і BackboneFast, протокол RSTP не є власністю однієї компанії.

Для забезпечення максимальної швидкості переконфігурації протокол RSTP вимагає повнодуплексного з'єднання "точка-точка" між комутаторами.

Переконфігурація зв'язного дерева при використанні протоколу RSTP займає менше однієї секунди, аналогічний процес протоколу STP займає близько однієї хвилини.

RSTP усуває потреби в таких функціях, як PortFast і UplinkFast. RSTP може переключатися в режим STP для обслуговування старого обладнання.

Для прискорення перерахунку число режимів портів протоколу RSTP зменшене до трьох: відхилення, навчання і пересилання. Режим відкидання аналогічний трьом оригінальним режимам STP: блокування, навчання і "відключений".

Крім того, у RSTP додана концепція активної топології. Усі порти, що не знаходяться в режимі відхилення, входять до складу активної топології і негайно переходять у режим пересилання.

Характеристики RSTP

RSTP забезпечує швидкий перерахунок зв'язного дерева: коли змінюється топологія мережі, RSTP може забезпечити збіжність в правильному налаштуванні мережі всього лише за кілька сотень мілісекунд. RSTP перевизначає тип портів та їх стан. Якщо порт налаштований як альтернативний або резервний порт він може відразу ж перейти в стан пересилання, не чекаючи збіжності мережі.

RSTP є найкращим протоколом для запобігання утворення петлі в комутованому мережевому середовищі. Такі удосконалення, як передавання в BPDU інформації про ролі портів тільки в сусідні комутатори, не вимагають додаткових налаштувань і в цілому виконується краще, ніж раніше в Cisco-пропрієтарних версіях. В даний час вони прозорі та інтегровані у функціонування протоколу.

Такі функції Cisco, як PortFast, UplinkFast і BackboneFast несумісні з протоколом RSTP.

В RSTP використовується поняття граничного порта по аналогії до механізму PortFast в протоколі STP. Це порт комутатора, який ніколи буде підключений до іншого комутатора. Такий порт після включення відразу ж переходить у стан пересилання.

Ні граничні порти, ні порти з підтримкою PortFast не спричиняють перегенерування змін в топології у випадку зміни свого стану.

На відміну від PortFast, граничний порт RSTP після отримання кадру BPDU втрачає статус граничного порту і негайно стає звичайним портом.

RSTP (802.1w) замінює STP (802.1D) при збереженні зворотньої сумісності. Основна частина термінології і більшість параметрів залишаються незмінними. Крім того, 802.1w здатен повертатись назад до 802.1D, для взаємодії з існуючими комутаторами. Наприклад, RSTP алгоритм вибирає кореневий міст аналогічно 802.1D.

RSTP (802.1w) використовує тип 2, версії 2 BPDU, завдяки чому RSTP міст може комунікувати з комутаторами, які використовують 802.1D. RSTP відправляє кадри BPDU і використовує поля flag в дещо інший спосіб, ніж STP.

Інформація протоколу визнається застарілою, якщо повідмлення-вітання не отримані протягом трьох послідовних діапазонів (протягом 6 секунд за замовчуванням, або якщо таймер max age завершився), що дозволяє швидко виявляти проблеми.

Стани портів та типи лінків в RSTP

Протокол RSTP забезпечує швидку збіжність при виникненні несправності або під час відновлення комутатора чи лінку. В протоколі RSTP роль порту відділена від стану порту. Наприклад, призначений порт може перебувати тимчасово в стані відхилення кадрів, хоча його кінцевим станом буде стан пересилання. В протоколі RSTP є три можливих стани портів: discarding (відхилення), learning (навчання) та forwarding (пересилання).

Тип лінку забезпечує класифікацію для кожного порту, який бере участь в RSTP. Тип зв'язку може визначати роль, яку відіграватиме порт при виконанні певних умов. Ці умови різні для прикордонних та неприкордонних портів. Неприкордонні порти поділяються на два типи лінків: точка-точка та спільні. Тип лінку визначається автоматично, але може бути переналаштований.

Прикордонні порти еквівалентні портам з підтримкою режиму PortFast, лінки типу точка-точка є кандидатами для швидкого переходу до стану пересилання. Однак, спочатку проткол RSTP повинен визначити роль портів, виходячи з того, що кореневі порти не використовують параметр "тип лінку". Кореневі порти можуть зробити швидкий перехід в стан пересилання, як тільки порт засинхронізується.

Альтернативні та резервні порти не використовують параметр "тип лінку" в більшості випадків.

Призначені порти використовують параметр «тип лінку». Швидкий перехід в режим пересилання для призначених портів відбувається тільки, якщо тип лінку «точка-точка».

Керування трафіком в корпоративних мережах

Ієрархічні корпоративні мережі спрощують обмін інформацією. Інформація циркулює між мобільними співробітниками і філіями, а філії підключаються до офісів компанії в містах і країнах усього світу. В організації повинна бути створена ієрархія, що відповідає різним мережевим вимогам того чи іншого підрозділу компанії.

Як правило, найважливіша інформація і служби розміщуються вгорі ієрархії на захищених серверних фермах або у мережах збереження даних. Структура розгортається в безліч різних відділів, які утворюють нижню частину ієрархії.

Для взаємодії між рівнями ієрархії необхідне поєднання технологій LAN і WAN.

У корпоративних мережах необхідне керування трафіком, інакше вони не зможуть функціонувати. Маршрутизатори направляють трафік і запобігають засміченню основних каналів важливих служб широкомовними розсиланнями. Вони керують потоками трафіку між LAN так, що через мережу надходить тільки потрібний трафік.

Корпоративні мережі передбачають високий рівень надійності та обслуговування. Для цього використовується ряд заходів:

– передбачаються резервні канали на випадок відмови основного маршруту;

– впроваджуються служби QoS, щоб важливі дані оброблялися в першу чергу;

– використовується фільтрація пакетів, щоб виключити деякі типи пакетів, збільшити пропускну здатність каналу і захистити мережі від загрози атак.

Корпоративні топології

Правильний вибір фізичної топології дозволяє компанії розширити свої мережеві службибез зниження їхньої надійності та продуктивності. Мережеві проектанти приймають рішення про вибір топології на основі корпоративних вимог до продуктивності і надійності. В корпоративних середовищах зазвичай впроваджуються топологія типу “зірка” і топологія сітки.

Топологія типу «зірка»

Одна з найбільш розповсюджених фізичних топологій - топологія типу “зірка”. Центр “зірки” відповідає вершині ієрархії, що може представляти головне управління чи головний офіс компанії. Філії в тих чи інших місцях розташування з’єднуються з центром зірки.

Топологія типу “зірка” забезпечує централізоване керування мережею. Усі важливі служби і технічний персонал можна розташувати в одному місці. Топології типу “зірка” можна масштабувати. При додаванні нової філії додається ще одне з’єднання з центральною точкою “зірки”. Якщо у відділення з’являються кілька філій у своєму місці розташування, кожне з них може з’єднатися з центральним вузлом відділення, який в свою чергу підключається до головної точки центрального офісу. У цьому випадку проста “зірка” може розростися до розширеної зірки з малими “зірками” у головних філіальних відділеннях.

Топології типу “зірка” і “розширена зірка” утворять єдину точку відмови. Коміркові топології дозволяють усунути цю проблему.

Коміркові топології (топології сітки)

Кожне додаткове з’єднання дає альтернативний канал передачі даних і підвищує надійність мережі. В міру додавання з’єднань топологія стає комірковою із взаємозв’язаними вузлами. Кожне додаткове з’єднання збільшує собівартість і накладні витрати. Більш того, при цьому ускладнюється керування мережами.

Частково-коміркова

З додаванням додаткових з’єднань тільки у визначені області корпоративної мережі утвориться частково-коміркова топологія. Ця топологія відповідає вимогам надійності і доступності в таких критично важливих областях, як серверні ферми і мережі збереження даних. Інші області мережі як і раніше піддані відмовам. Таким чином, коміркову топологію необхідно розміщувати там, де це найбільш вигідно.

Повнозв’язна топологія

Якщо простої в роботі мережі недопустимі, тоді потрібна повнозв’язна топологія. У повнозв’язній топології кожен вузол з’єднується з усіма вузлами в компанії. Це найбільш відмовостійка топологія, але її впровадження вимагає найбільших витрат.

Мережа Інтернет – яскравий приклад коміркової топології. Керування пристроями в мережі Інтернет виконується не однією людиною або організацією. У результаті топологія мережі Інтернет постійно змінюється – деякі з'єднання стають активними, а інші неактивними. Додаткові з'єднання дозволяють збалансувати трафік і забезпечують надійний канал до адреси призначення.

Деякі з проблем мережі Інтернет постають і перед корпоративними мережами. Тому передбачені визначені процедури, що дозволяють пристроям адаптуватися до цих неперервно змінних умов і належним чином направляти трафік.

Статична та динамічна маршрутизація

Фізична топологія корпоративної мережі задає структуру для передачі даних. Маршрутизація забезпечує механізм її реалізації. У корпоративних мережах ускладнюється пошук оптимального маршруту до адреси призначення, оскільки в маршрутизатора може бути багато джерел інформації, на основі якої створюється таблиця маршрутизації.

Таблиця маршрутизації [11, 12] – це файл даних, що знаходиться в ОЗП і зберігає дані про підключення віддалених та безпосередньо під'єднаних мереж. У таблиці маршрутизації кожна мережа зв'язана або з вихідним інтерфейсом, або з наступним переходом.

Вихідний інтерфейс – це фізичний шлях, що використовується маршрутизатором для переміщення даних ближче до адреси призначення. Наступний перехід – це інтерфейс підключеного маршрутизатора, що переміщує дані ближче до адреси кінцевого призначення.

Крім того, у таблиці кожному маршруту призначається номер, що відображає ймовірність і точність джерела даних про маршрутизацію – адміністративна відстань. Маршрутизатори обслуговують дані про безпосередньо підключені, статичні та динамічні маршрути.

Маршрути з прямим підключенням

Безпосередньо підключена мережа підключається до інтерфейсу маршрутизатора. За допомогою налаштування інтерфейсу з IP-адресою та маскою інтерфейс стає вузлом у підключеній мережі. Адреса мережі і маска інтерфейсу разом з типом і номером інтерфейсу відображаються в таблиці маршрутизації як безпосередньо підключена мережа. У таблиці маршрутизації безпосередньо підключені мережі позначаються символом C.

Статичні маршрути – це маршрути, що налаштовуються адміністратором мережі вручну. Статичний маршрут містить у собі адресу мережі і маску для мережі призначення разом з вихідним інтерфейсом або IP-адресою маршрутизатора наступного переходу. У таблиці маршрутизації статичні маршрути позначаються символом S. У статичних маршрутів найменша адміністративна відстань, оскільки вони стабільніші та надійніші за динамічні маршрути.

Протоколи динамічної маршрутизації також додають віддалені мережі в таблицю маршрутизації. Вони дозволяють маршрутизаторам спільно використовувати інформацію про надійність і статус віддалених мереж за допомогою виявлення мережі. Кожен протокол відправляє та отримує пакети даних, виконуючи пошук інших маршрутизаторів, оновлюючи та обслуговуючи таблиці маршрутизації. Маршрути, отримані за допомогою протоколів динамічної маршрутизації, визначаються протоколом. Наприклад, R позначається протокол RIP, а D – протокол EIGRP. Їм призначається адміністративна відстань протоколу.

Як правило, в корпоративній мережі використовуються і статичні, і динамічні маршрути. Статична маршрутизація спрямована на рішення конкретних мережевих задач. У залежності від фізичної топології за допомогою статичного маршруту можна керувати потоками трафіку.

Якщо обмежити трафік однією точкою входу/виходу, буде створена закрита мережа. У філіях деяких корпоративних мереж є тільки один можливий маршрут до іншої частини мережі. У цьому випадку кінцевий маршрутизатор не буде обтяжений відновленнями маршрутів і збільшенням навантаження через виконання протоколу динамічної маршрутизації, тому статична маршрутизація більш виграшна.

У залежності від розташування і функцій для деяких корпоративних маршрутизаторів можливе виникнення потреби використання статичних маршрутів. Прикордонні маршрутизатори використовують статичні маршрути для забезпечення безпечних стабільних

маршрутів до ISP. Інші маршрутизатори використовують протоколи статичної, або динамічної маршрутизації відповідно до задач.

Маршрутизатори корпоративної мережі використовують пропускну здатність, пам'ять і обчислювальні ресурси для перетворення NAT/PAT, фільтрації пакетів і інших сервісів. Статична маршрутизація дозволяє виконувати пересилання, уникаючи навантаження, що пов'язане з більшістю протоколів динамічної маршрутизації.

Статична маршрутизація передбачає вищий рівень безпеки, ніж динамічна, оскільки не вимагає відновлення маршрутів. Хакер може перехопити відновлення динамічної маршрутизації, щоб отримати дані про мережу.

Проте, і при статичній маршрутизації можуть виникнути проблеми. Вона вимагає тимчасових витрат і точності з боку мережевого адміністратора, що змушений вручну вводити дані маршрутизації. Проста помилка в статичному маршруті може привести до простою мережі та втрати пакетів. Після зміни статичних маршрутів у мережі можуть виникнути помилки і збої маршрутизації в ході ручного переналаштування. З цих причин статична маршрутизація не підходить для повсякденного використання у великих корпоративних середовищах.

Налаштування статичних маршрутів

Глобальною командою для налаштування більшості статичних маршрутів є ip route з вказанням мережі призначення, маски та шляху до неї. Таким чином, команда наступна:

Router(config)#ip route [адреса мережі] [маска підмережі] [адреса наступного переходу або вихідного інтерфейсу]

За допомогою адреси наступного переходу, або вихідного інтерфейсу, маршрутизатор направляє трафік до потрібної адреси призначення. Однак ці два параметри діють по-різному.

Перед пересиланням маршрутизатором пакету процес у таблиці маршрутизації визначає вихідний інтерфейс для використання. Пошук у таблиці маршрутизації по статичних маршрутах, налаштованих для роботи з вихідними інтерфейсами, здійснюється лише один раз. Тоді як статичним маршрутам з налаштованим параметром наступного переходу доводиться звертатися до таблиці маршрутизації двічі, щоб визначити вихідний інтерфейс.

У корпоративній мережі статичні маршрути, налаштовані для роботи з вихідними інтерфейсами, ідеальні для з'єднань точка-точка, наприклад, для з'єднань між прикордонним маршрутизатором та ISP.

Статичним маршрутам, налаштованим для роботи з інтерфейсом наступного переходу, потрібно два кроки, щоб визначити вихідний інтерфейс. Це і є рекурсивний пошук. У ході рекурсивного пошуку:

– маршрутизатор співставляє IP-адресу призначення для пакету зі статичним маршрутом;

– далі він співставляє IP-адресу наступного переходу статичного маршруту з записами в таблиці маршрутизації, щоб визначити інтерфейс для використання.

Якщо відключений вихідний інтерфейс, статичні маршрути не будуть відображатися в таблиці маршрутизації. Після включення інтерфейсу маршрути будуть у ній перевстановлені.

Об'єднання декількох статичних маршрутів в один запис скорочує розмір таблиці маршрутизації і підвищує ефективність процесу пошуку. Цей процес називається сумуванням маршрутів.

Один статичний маршрут підсумовує кілька статичних маршрутів, якщо:

– мережі призначення об'єднані в єдину мережеву адресу;

– усі статичні маршрути використовують однакову IP-адресу вихідного інтерфейсу або наступного переходу.

Без сумарних маршрутів таблиці маршрутизації на магістральних маршрутизаторах мережі Інтернет стають некерованими. У корпоративних мережах виникають ті ж проблеми. Сумарні статичні маршрути – незамінне рішення в керуванні розмірами таблиць маршрутизації.

В залежності від корпоративних служб WAN статичні маршрути можуть забезпечувати резервне копіювання при відмові з'єднання основної WAN. У цьому випадку з метою резервного копіювання використовується функція плаваючих статичних маршрутів.

За замовчуванням адміністративна відстань статичного маршруту менша за адміністративну відстань маршруту, отриманого з використанням протоколів динамічної маршрутизації. Адміністративна відстань плаваючого статичного маршруту більша за адміністративну відстань маршруту, отриманого по протоколу динамічної маршрутизації. З цієї причини плаваючий статичний маршрут не відображається в таблиці маршрутизації. Запис плаваючого статичного маршруту буде відображений в таблиці маршрутизації, тільки якщо дані динамічних протоколів будуть втрачені.

Для створення плаваючого статичного маршруту потрібно додати значення для адміністративної відстані в кінці команди ip route:

Router(config)#ip route 192.168.4.0 255.255.255.0 192.168.9.1 200

Адміністративна відстань повинна бути більша AD, призначеної протоколу динамічної маршрутизації. Маршрутизатор використовує основний маршрут, поки він активний. Якщо основний маршрут стає неактивним, у таблиці буде встановлений плаваючий статичний маршрут.

Маршрути за замовчуванням

У таблицях маршрутизації не може бути маршрутів для всіх можливих вузлів мережі Інтернет. Зростання розмірів таблиць маршрутизації потребує більше ОЗП та обчислювальної потужності. Спеціальний тип статичного маршруту, названий маршрутом за замовчуванням, вказує який шлюз використовувати, якщо в таблиці маршрутизації немає шляху до адреси призначення. Звичайно маршрути за замовчуванням вказують наступний маршрутизатор на шляху до ISP. У складних корпоративних середовищах маршрути за замовчуванням виводять Інтернет-трафік з мережі.

Команда для створення маршруту за замовчуванням схожа з командою для створення звичайного чи плаваючого статичного маршруту. Мережева адреса і маска позначаються як 0.0.0.0, в результаті отримуємо маршрут чотирьох нулів. У команді використовується або адреса наступного переходу, або параметри вихідного інтерфейсу.

Нулі вказують маршрутизатору, що для використання цього маршруту біти збігатися не повинні. Якщо не існує більш оптимального маршруту, маршрутизатор буде використовувати статичний маршрут за замовчуванням.

Кінцевий маршрут за замовчуванням, розташований на пограничному маршрутизаторі, відправляє трафік до ISP. Цей маршрут позначає останній вузол в корпоративній мережі, як шлюз "останньої надії" для пакетів, що не вдається доставити. Ці дані відображаються в таблицях маршрутизації всіх маршрутизаторів.

Якщо в корпоративній мережі використовується протокол динамічної маршрутизації, пограничний маршрутизатор може відправляти маршрут за замовчуванням іншим маршрутизаторам у формі відновлення динамічних маршрутів.

Призначення протоколів динамічної маршрутизації

Протоколи маршрутизації використовуються для полегшення обміну інформацією про маршрутизацію між маршрутизаторами. Протоколи маршрутизації дозволяють маршрутизаторам динамічно обмінюватися інформацією про віддалені мережі і автоматично додавати цю інформацію в своїх таблицях маршрутизації.

Протоколи маршрутизації визначають найкращий шлях до кожної мережі, який потім додається в таблицю маршрутизації. Одним з основних переваг використання протоколів динамічної маршрутизації є те, що маршрутизатори обмінюються маршрутною інформацією після зміни топології. Такий обмін дозволяє маршрутизаторам автоматично дізнаватися про нові мережі, а також знайти альтернативні шляхи, коли виникають проблеми з існуючим маршрутом.

В порівнянні зі статичною маршрутизацією, протоколи динамічної маршрутизації, вимагають менше затрат на адміністрування. Але відбувається це за рахунок використання частини ресурсів маршрутизатора для забезпечення роботи протоколів роботи (процесор, оперативна пам'ять) та пропускної здатності каналу зв'язку. Незважаючи на переваги

динамічної маршрутизації, статична маршрутизація досі займає важливе місце в маршрутизації. Є моменти, коли статична маршрутизація є доцільнішою, хоча найчастіше ці типи маршрутизації поєднуються.

Протокол маршрутизації – це набір процесів, алгоритмів та повідомлень, які використовуються для обміну інформацією про маршрутизацію та заповнення таблиці маршрутизації та вибору найкращого маршруту. Протоколи маршрутизації забезпечують:

– виявлення віддалених мереж;

– підтримку актуальної інформації;

– вибір найкращого шляху до мережі призначення;

– можливість знайти новий кращий шлях, якщо поточний шлях більше недоступний.

Компонентами протоколу маршрутизації є:

– структури даних – деякі протоколи маршрутизації використовують таблиці та/або бази даних для своїх операцій. Ця інформація зберігається в оперативній пам'яті;

– алгоритм – це набір кроків, що використовується протоколами маршрутизації, обробки інформації про маршрутизацію та визначення найкращого маршруту;

– повідомлення – протоколи маршрутизації використовують різні типи повідомлень для виявлення сусідніх маршрутизаторів, обміну інформацією про маршрутизацію, а також інші завдання для вивчення та підтримки іформації про мережу.

Всі протоколи маршрутизації мають однакову мету – дізнаватися про віддалені мережі та швидко адаптуватися до змін в топології. Методи, які використовують протоколи маршрутизації для досягнення цієї мети залежать від алгоритму, який використовується протоколом та характеристик самого протоколу.

Класифікація протоколів динамічної маршрутизації

Протоколи маршрутизації поділяються на групи в залежності від характеристик. Найбільш вживаними протоколами маршрутизації є:

RIP – дистанційно-векторний протокол внутрішньої маршрутизації.

IGRP – дистанційно-векторний протокол розроблений Cisco (застарілий на сьогоднішній день).

OSPF – внутрішній протокол стану каналу.

IS-IS – внутрішній протокол стану каналу.

EIGRP – покращений дистанційно-векторний протокол, розроблений Cisco.

BGP – протокол вектору шляху зовнішньої маршрутизації.

Протоколи IGP та EGP

Автономна система (AS - autonomous system), або домен маршрутизації – це набір маршрутизаторів під спільним адмініструванням. Типовими прикладами є внутрішні мережі компаній та мережі Інтернет-провайдерів. Оскільки Інтернет заснований на концепції автономної системи потрібні два типи протоколів маршрутизації: внутрішні та зовнішні. Цими протоколами є:

Внутрішні протоколи (IGP - Interior Gateway Protocols) використовуються для маршрутизації всередині автономної системи.

Зовнішні протоколи (EGP - Exterior Gateway Protocols) використовуються для міждоменної маршрутизації – маршрутизації між автономними системами.

Характеристики IGP та EGP протоколів маршрутизації

Протоколи внутрішньої маршрутизації використовуються в межах домену, який контролюється однією організацією. Автономна система зазвичай складається з безлічі окремих мереж, що належать компаніям, закладам, установам. Протоколи IGP використовуються для маршрутизації в межах автономної системи, а також для маршрутизації в межах окремих мереж. До протоколів IGP належать RIP, IGRP, EIGRP, OSPF та IS-IS.

Протоколи EGP призначені для використання між різними автономними системами, які знаходяться під контролем різних адміністрацій. На даний час BGP є єдиним життєздатним EGP протоколом маршрутизації, який використовується в Інтернет. BGP є протоколом вектора шляху та може використовувати багато атрибутів для оцінки маршрутів. На рівні провайдера, часто є більш важливі питання, ніж просто вибір найшвидшого

маршруту. BGP зазвичай використовується між провайдерами, а іноді і між компанією та Інтернет-провайдером.

Конвергенція (або збіжність) – це стан мережі, коли таблиці маршрутизації всіх маршрутизаторів знаходяться в стані узгодженості. Мережа конвергентна тоді, коли всі маршрутизатори мають повну і точну інформацію про мережу. Час збіжності – це час, який необхідний маршрутизаторам для обміну інформацією, обчислення найкращих маршрутів та оновлення таблиць маршрутизації. Мережа не є повноцінно робочою, поки не відбудеться конвергенція. Через це більшість мереж вимагають короткий час конвергенції.

Конвергенція характеризується швидкістю поширення маршрутної інформації та розрахунку оптимальних маршрутів. Протоколи маршрутизації можуть порівнюватись на основі швидкості збіжності. Швидша конвергенції характеризує кращий протокол маршрутизації.

Метрики

Є випадки, коли протокол маршрутизації дізнається про більш, ніж один маршрут до певного місця призначення. Щоб вибрати кращий шлях, протокол маршрутизації повинен вміти оцінити відмінності між доступними шляхами. Для цього використовується метрика. Метрика – це значення, яке використовується протоколами маршрутизації для визначення вартості доступу до віддалених мереж.

Різні протоколи маршрутизації використовують різні метрики. Метрики, що використовуються одним протоколом, відрізняються від метрик іншого протоколу. Два різних протоколи маршрутизації виберуть різні маршрути до одніє мережі призначення за рахунок використання різних метрик.

RIP буде вибирати шлях з найменшою кількістю стрибків, в той час як OSPF вибиратиме шлях з високою пропускною здатністю.

Протоколи маршрутизації використовують такі метрики:

Hop count (кількість переходів) – проста метрика, яка підраховує кількість маршрутизаторів, через які повинен пройти пакет.

Bandwidth (смуга пропускання) – вибір маршруту з врахуванням найвищої пропускної здатності.

Load (завантаження) – враховує завантаженість каналів передачі.

Delay (затримка) – враховує час проходження пакета по маршруту.

Reliability (надійність) – оцінює ймовірність збою каналу передачі, розраховується на основі помилок інтерфейсу або попередніх збоїв.

Cost (вартість) – вартість, визначається IOS або адміністратором мережі, щоб вказати перевагу маршруту. Вартість може представляти метрики, поєднання метрик чи політики.

Метриками для різних протоколів є:

RIP: кількість переходів – найкращим шляхом вибирається шлях з найменшою кількістю переходів.

IGRP та EIGRP: пропускна здатність, затримка, надійність та завантаження – найкращим шляхом вважається маршрут з найменшою композитною метрикою, обчисленою з врахуванням цих параметрів. За замовчуванням використовується лише пропускна здатність та затримка.

IS-IS та OSPF: вартість – найкращим шляхом вибирається шлях з найменшою вартістю. Варіант реалізації OSPF компанією Cisco використовує пропускну здатність.

Балансування навантаження (Load balancing)

Протоколи використовують метрики для визначення найкращого шляху до віддаленої мережі. Але можлива ситуація, коли кілька маршрутів мають однакові значення метрики. В такому випадку маршрутизатор вибирає не один маршрут, а балансує навантаження між цими однаковими маршрутами. Пакети пересилаються з використанням рівноцінних маршрутів.

Протоколи динамічної маршрутизації поділяються на дві основні категорії: «вектор відстані» та «стану каналу».

Протоколи маршрутизації типу «вектор відстані»

Вектор відстані означає, що маршрути визначаються як вектори відстані та напрямку. Відстань визначається в термінах метрики, наприклад, як число переходів, а напрям – це маршрутизатор наступного пересилання або вихідний інтерфейс.

Протоколи векторів відстані зазвичай використовують алгоритм Беллмана-Форда для визначення кращого маршруту.

Деякі протоколи векторів відстані періодично відправляють повні таблиці маршрутизації до всіх підключених сусідів. У великих мережах такі оновлення маршрутів можуть створювати значні об'єми трафіку.

Хоча алгоритм Беллмана-Форда в кінцевому рахунку накопичує достатньо інформації, для підтримки бази даних досяжності мереж, алгоритм не дозволяє маршрутизатору знати топологію всієї мережі. Маршрутизатор знає тільки інформацію отриману від своїх сусідів.

Єдиною інформацією, якою володіє маршрутизатор про віддалену мережу, є відстань або метрика для досягнення цієї мережі, та інформація про те, який шлях або інтерфейс використовувати, щоб туди дістатися. Протоколи векторів відстані не мають фактичної карти топології мережі.

Протоколи векторів відстані працюють найкраще в таких умовах:

– мережа є простою та плоскою і не потребує спеціального ієрархічного дизайну;

– мережеві адміністратори не володіють достатніми знаннями для налаштування та діагностики протоколів стану каналу;

– в певних типах мереж, таких як зіркоподібні мережі (hub-and-spoke networks);

– час конвергенції мережі не має значення.

Динамічні протоколи маршрутизації допомагають мережевому адміністратору зекономити час на налаштування та обслуговування статичних маршрутів.

До протоколів векторів відстані належать протоколи:

– Routing Information Protocol (RIP);

– Interior Gateway Routing Protocol (IGRP);

– Enhanced Interior Gateway Routing Protocol (EIGRP).

RIP

Протокол RIP був описаний в RFC 1058. Він має такі основні характеристики:

– лічильник переходів використовується як метрика для вибору маршруту;

– якщо кількість переходів більше 15, RIP не може забезпечити маршрут до цієї мережі;

– оновлення інформації про маршрутизацію відбувається шляхом широкомовної або групової розсилки кожні 30 секунд, за замовчуванням.

IGRP

Протокол IGRP є власною розробкою компанії Cisco. IGRP має такі основні характеристики дизайну:

– пропускна здатність, затримка, надійність та завантаження використовуються для створення композитної метрики;

– оновлення інформації про маршрутизацію відбувається шляхом широкомовної розсилки кожні 90 секунд, за замовчуванням;

– IGRP є попередником EIGRP і на даний час не використовується.

EIGRP

Протокод EIGRP є власністю Cisco і має такі основні характеристики:

– він може виконувати балансування навантаження;

– він використовує алгоритму DUAL (Diffusing Update Algorithm) для розрахунку найкоротшого шляху;

– в ньому не використовуються періодичні оновлення. Оновлення маршрутної інформації здійснюється тільки тоді, коли відбуваються зміни в топології.

Як і всі протоколи маршрутизації, протоколи на основі векторів відстані використовують метрику для визначення оптимального маршруту. Протоколи на основі

векторів відстані розраховують оптимальний маршрут, виходячи з відстані від маршрутизатора до мережі.

Протоколам, на основі векторів відстані, звичайно потрібно менш складне налаштування і керування в порівнянні з протоколами на основі стану каналу. Вони можуть виконуватися маршрутизаторами старіших моделей з меншою потужністю і вимагають меншого об'єму пам'яті та обчислень.

Маршрутизатори, що використовують протоколи на основі векторів відстані, виконують широкомовну чи багатоадресну розсилку всієї таблиці маршрутизації своїм сусідам через рівні інтервали часу. Якщо маршрутизатор отримує більше одного маршруту до адреси призначення, він розраховує і передає маршрут з найменшою метрикою.

Цей спосіб передачі даних маршрутизації у великих мережах відрізняється малою швидкістю. У визначений момент у деяких маршрутизаторів може не бути останніх відомостей про мережу. Це обмежує масштабованість протоколів і викликає проблеми, наприклад, петлі маршрутизації.

Періодичні оновлення та підтримка таблиць маршрутизації

Багато протоколів векторів відстані використовують періодичні оновлення для обміну маршрутною інформацією з сусідами та підтримування актуальної інформації в таблиці маршрутизації. Протоколи динамічної маршрутизації RIP та IGRP є прикладами таких протоколів.

Таймер оновлення інформації про маршрутизацію в таблиці маршрутизації оновлюється щоразу після отримання оновлення. Таким чином, інформація в таблиці маршрутизації може бути змінена при змінах в топології. Зміни можуть відбуватися з кількох причин, серед яких:

– відмова каналу зв'язку;

– додавання нового каналу зв'язку;

– відмова маршрутизатора;

– зміна параметрів каналу зв'язку.

Петля маршрутизації – це ситуація, при виникненні якої пакет постійно передається між маршрутизаторами і ніколи не досягає своєї мережі призначення. Петля маршрутизації може виникнути у випадку, коли два або більше маршрутизаторів мають інформацію про маршрутизації, яка вказує, що існує шлях до недосяжної мережі.

Петля може виникнути в результаті:

– неправильно налаштованих статичних маршрутів;

– неправильно налаштованого перерозподілу маршрутів (route redistribution);

– непослідовності таблиць маршрутизації, яка виникає через повільну збіжність мережі.

Простота роботи протоколів типу вектора відстані є причиною такого недоліку, як петлі маршрутизації, хоча за певних умов петлі маршрутизації можуть виникати і в протоколах типу стану каналу.

IP-протокол має свій власний механізм для запобігання нескінченній передачі пакетів через мережу. Для цього використовується поле TTL (Time-to-Live). Його значення зменшується на одиницю при проходженні через кожен маршрутизатор на шляху проходження від відправника до одержувача. Якщо TTL дорівнює нулю, маршрутизатор відкидає пакет.

Виникнення петель маршрутизації спричинює зниження продуктивності або, навіть, простій мережі.

Петлі маршрутизації виникають за певних умов:

– пропускна здатність каналу зв'язку буде використовуватися для передачі;

– процесор маршрутизатора буде завантажений циклічними пакетами;

– процесор маршрутизатора буде завантажений пересиланням непотрібних пакетів, що в свою чергу вплине на час збіжності мережі;

– оновлення маршрутизації можуть втрачатись або не оброблятись вчасно. Це створюватиме нові петлі маршрутизації, погіршуючи ситуацію загалом.

Для усунення петель маршрутизації використовується ряд механізмів, в першу чергу в протоколах вектора відстані. До цих механізмів належать:

– визначення максимальної метрики для запобігання нескінченної передачі;

– holddown timers (таймери утримання);

– split horizon(розділення горизонту);

– route poisoning or poison reverse (зворотнє виправлення);

– triggered updates (миттєве оновлення).

Встановлення максимальної метрики дозволяє після певної кількості переходів позначити мережу недосяжною та відкинути пакет.

Таймери утримання використовуються для запобігання оновленням з неналежно відновлених чи поганих маршрутів. Таймери визначають період, протягом якого маршрутизатори не проводять будь-які зміни. Якщо маршрут позначається як втрачений, або, як можливо втрачений, будь-яка інша інформація про цей маршрут, яка містить такий же статус, або гірший статус, ігнорується протягом певного періоду часу (період утримання). Це означає, що маршрутизатори залишають маршрут в таблиці маршрутизації позначеним як недосяжний протягом часу, достатнього для розповсюдження таблиць маршрутизації з останньою актуальною інформацією до всіх маршрутизаторів в мережі.

Правило розділення горизонту визначає, що маршрутизатор не повинен надсилати оновлення про мережу через інтерфейс, з якого надійшли оновлення. Split horizon може бути відключений адміністратором.

Route poisoning використовується для позначення маршруту, як недосяжного в оновленнях, які надсилаються на інші маршрутизатори. Недосяжність інтерпретується як метрика, яка встановлюється на максимум. Для RIP poisoned route має метрику 16.

Щоб пришвидшити процес конвергенції після зміни топології, RIP використовує миттєві оновлення. Миттєві оновлення оновлюють таблиці маршрутизації негайно у відповідь на зміну маршруту. Миттєві оновлення не чекають завершення таймера оновлення. Отримавши оновлення, маршрутизатор негайно відправляє повідомлення про оновлення на сусідні маршрутизатори.

Миттєві оновлення надсилаються у випадку:

– зміни стану інтерфейсу (up або down);

– маршрут став “недосяжним”;

– маршрут додано в таблицю маршрутизації;

Використання миттєвих оновлень було б достатньо, якщо б існувала гарантія, що хвиля оновлення досягне всіх необхідних маршрутизаторів негайно. Проте, є дві проблеми з миттєвими оновленнями:

– пакети, що містять оновлення можуть бути видалені або пошкоджені;

– миттєві оновлення не відбуваються миттєво. Цілком можливо, що маршрутизатор, який ще не отримав миттєве оновлення надсилатиме регулярне оновлення у невідповідний час, що спричинить встановлення поганого маршруту в сусідньому маршрутизаторі, який вже отримав миттєве оновлення.

Критерії вибору протоколу

Приймаючи рішення, який протокол вектора відстані вибрати, необхідно враховувати декілька факторів:

– розмір мережі

– сумісність між моделями маршрутизаторів

– знання та вміння адміністратора

Протоколи RIP версії 1 і 2 є звичайними протоколами на основі векторів відстані, а протокол EIGRP – протоколом на основі векторів відстані з розширеними можливостями. Протокол RiPng, нова версія протоколу RIP, був спеціально розроблений для підтримки IPv6.

Протягом багатьох років протокол RIP перетворився з протоколу повнокласової маршрутизації (RIPv1) на протокол безкласової маршрутизації (RIPv2). RIPv2 – це стандартизований протокол маршрутизації, що працює на обладнанні різних виробників. Це один з найпростіших протоколів маршрутизації для налаштування, що робить його гарним

вибором для невеликих мереж. Проте RIPv2 також має певні обмеження. Протоколи RIPv1 та RIPv2 використовують кількість переходів в якості метрики маршруту, з максимальним обмеженням в 15 переходів, що обмежує їх використання.

Особливості RIP:

– підтримуєм split horizon та split horizon з poison reverse для запобігання утворенню петель;

– забезпечує балансування навантаження з підтримкою до шести рівноцінних по вартості маршрутів. За замовчуванням використовується чотири рівноцінні маршрути.

В RIPv2 внесені такі вдосконалення:

– в оновлення включається маска підмережі;

– механізм аутентифікації для оновлення таблиці маршрутизації;

– підтримка маски змінної довжини (VLSM);

– використання групових адрес замість широкомовних;

– підтримка сумування маршруту.

Протокол EIGRP (Enhanced IGRP) був розроблений на базі протоколу IGRP. EIGRP безкласовий протокол типу вектора відстані з певними функціями схожими на функції протоколів стану каналу. Це є пропрієтарний протокол і працює лише на маршрутизаторах Cisco.

Основні характеристики EIGRP:

– миттєві оновлення (EIGRP не використовує періодичних оновлень);

– використовує таблицю топології для підтримки всіх маршрутів, отриманих від сусідів (не тільки кращих шляхів);

– створення суміжніх з'єднань з сусідніми маршрутизаторами, які використовують протокол EIGRP hello;

– підтримка VLSM та ручного сумування маршрутів.

Переваги EIGRP:

– хоча маршрути поширюються за принципом вектора відстані, метрика базується на мінімальній ширині смуги та затримці маршруту, а не кількості стрибків;

– швидка збіжність завдяки використанню алгоритму DUAL(Diffusing Update Algorithm) для обчислення маршруту. DUAL дозволяє вставляти в таблиці EIGRP топології резервні маршрути, які використовуються у випадку, якщо основний маршрут пошкоджується. Завдяки цьому перемикання на резервний маршрут відбувається негайно і не потребує жодних дій від будь-яких інших маршрутизаторів.

– обмежені оновлення означають, що EIGRP використовує менше пропускної здатності, особливо у великих мережах з великою кількістю маршрутів;

– EIGRP підтримує декілька протоколів мережевого рівня завдяки використанню незалежних модулів, які включають в себе підтримку IP, IPX та AppleTalk.

Характеристики протоколу RiPv1

Протокол RIP був першим протоколом маршрутизації на основі вектору відстані IP, стандартизованого у RFC (RFC1058 у 1988 році). Першу версію RIP тепер часто називають RiPv1, щоб відрізняти її від згодом удосконаленої версії RiPv2, а також від версії IPv6 – RiPng.

Основні характеристики RIPv1

– повнокласовий протокол типу вектора відстані;

– в ролі метрики використовує кількість переходів;

– маршрути, з кількістю переходів понад 15, недосяжні;

– широкомовна розсилка оновлень кожні 30 секунд.

Дані RIP повідомлення інкапсулюються в UDP сегмент з номером порту відправника та отримувача 520. В заголовок IP та заголовок канального рівня додається широкомовна адреса призначення перед надсиланням через усі налаштовані інтерфейси.

RIP таймери

За замовчуванням протокол RiPv1 виконує широкомовну передачу оновлень маршрутів по всіх активних інтерфейсах кожні 30 секунд.

Крім таймеру оновлення, використовуються ще три таймери:

– Invalid

– Flush

– Holddown

Invalid timer. Якщо оновлення для існуючого маршруту не були отримані протягом 180 секунд (за замовчуванням), маршрут позначається як недійсний, і йому встановлюється метрика 16. Маршрут залишається в таблиці маршрутизації до моменту завершення flush таймера.

Flush timer. За замовчуванням, таймер встановлюється на 240 секунд, тобто на 60 секунд довше, ніж invalid таймер. Коли flush таймер завершується, маршрут видаляється з таблиці маршрутизації.

Holddown timer. Цей таймер стабілізує маршрутну інформацію та допомагає запобігати петлям маршрутизації в періоди здійснення конвергенції мережі. Як тільки маршрут позначається як недосяжний, він залишається в стані утримання протягом часу, достатнього для отримання усіма маршрутизаторами інформації про недоступність мережі. За замовчуванням таймер встановлюється на 180 секунд..

RiPv1 є протоколом повнокласової маршрутизації. Він автоматично підсумовує підмережі по класовій межі і не відправляє, у оновленні, дані про мережеву маску. Відповідно, RiPv1 не підтримує VLSM і CIDR. Маршрутизатор, що працює по протоколу RiPv1, або використовує задану для локального інтерфейсу мережеву маску, або застосовує мережеву маску на основі класу адреси, яка використовується по замовчуванню. Через це обмеження підмережі, що повідомляються протоколом RiPv1, не можуть бути несуміжними.

Наприклад, маршрутизатор із заданими інтерфейсами, як шлюзами для підмереж 172.16.1.0/24 і 172.16.4.0/24, при використанні протоколу RiPv1 буде повідомляти тільки мережу класу B 172.16.0.0. Відповідно, інший маршрутизатор, що отримує це поновлення, буде вказувати мережу 172.16.0.0 у своїй таблиці маршрутизації. Це означає, що пакети з фактичною адресою підмережі призначення 172.16.3.0 можуть бути помилково спрямовані на інший маршрутизатор і не прибудуть у потрібну підмережу призначення.

За замовчуванням RIP має адміністративну відстань 120.

Зупинка непотрібних оновлень RIP протоколу.

Для запобігання передачі оновлень через певні інтерфейси маршрутизатора використовується команда:

Router(config-router)#passive-interface interface-type interface-number

Ця команда зупиняє оновлення маршрутів із зазначеного інтерфейсу. Проте, мережа, до якої належить вказаний інтерфейс буде передаватись в оновленнях маршрутів, які відправляються з інших інтерфейсів.

Автоматичне сумування машрутів

Протокол RIP є повнокласовим протоколом, який автоматично сумує повнокласові мережі на кордоні мережі.

Правила обробки оновлень:

– якщо оновлення та інтерфейс, який їх приймає, належать до однієї і тієї ж мережі, тоді мережева маска інтерфейсу застосовується до мережі в оновленні

– якщо оновлення та інтерфейс, який їх приймає, належать до різних мереж, тоді повнокласова маска застосовується до мережі в оновленні

Переваги автоматичного сумування маршрутів.

Протокол RIP автоматично сумує повно класові мережі. Оскільки оновлення 172.30.0.0 надсилається через інтерфейс (Serial 0/0/1) в іншу повнокласову мережу (192.168.4.0), RIP надсилає повнокласову мережу замість окремих підмереж.

Це зменшує розмір оновлень та завантаженість каналів зв’язку (між R2 та R3 в даному прикладі).

R3 отримає один маршрут для мережі 172.30.0.0/16 незалежно від того, скільки є підмереж. Використання єдиного маршруту пришвидшує пошук в таблиці маршрутизації маршрутизатора R3.

Недоліки автоматичного сумування

Протоколи повнокласової маршрутизації не включають мережевої маски в оновлення. Мережі автоматично сумуються на границі мережі.

Наприклад маршрутизатори R1 та R3 мають підмережі з мережі 172.30.0.0/16, в той час як R2 не має. Отже, R1 та R3 є прикордонними маршрутизаторами для 172.30.0.0/16, оскільки вони розділені іншою мережею 209.165.200.0/24. Це розділення створює несуміжні мережі (discontiguous network), у вигляді двох груп підмереж 172.30.0.0/24, розділених іншою мережею. 172.30.0.0/16 є несуміжною мережею.

Навіть за умови правильного налаштування RIPv1 він не здатний визначити всі мережі в несуміжній топології. Так R1 не буде оголошувати 172.30.1.0 або 172.30.2.0 до R2 через мережу 209.165.200.0. Аналогічно R3 не буде оголошувати 172.30.100.0 або 172.30.200.0 до R2 через мережу 209.165.200.0. Хоча обидва маршрутизатори R1 та R3 будуть оголошувати 172.30.0.0.

Характеристики протоколу RiPv2

Протокол RIPv2 описаний в RFC 1723. Подібно до RIPv1, RIPv2 інкапсулюється в UDP сегмент використовуючи порт 520 і може переносити інформацію про 25 маршрутів. RIPv2 має такий самий формат повідомлення як RIPv1, але додаються два суттєві розширення.

Першим розширенням є поле мережевої маски, що дозволяє включати 32 бітну маску в запис маршруту. Завдяки цьому більше не існує залежності мережевої маски від вхідного інтерфейсу чи повно класової маски при визначенні маски маршруту.

Другим суттєвим розширенням формату повідомлення RIPv2 є додавання адреси наступного переходу (Next Hop Address). Ця адреса використовується для визначення кращої адреси наступного переходу ніж адреса маршрутизатора, який надсилає повідомлення. Якщо всі поля встановлені в нулі (0.0.0.0), адреса маршрутизатора, який надсилає повідомлення буде найкращою адресою наступного переходу.

У протоколу RiPv2 багато функцій, схожих з RiPv1. Крім того, він передбачає важливі удосконалення. RiPv2 – це протокол безкласової маршрутизації, що підтримує VLSM і CIDR. Поле маски включено в оновлення версії 2, що дозволяє використовувати несуміжні мережі. Крім того, протокол RiPv2 дає можливість відключити автоматичне підсумовування маршрутів.

Обидві версії протоколу RIP розсилають усю таблицю маршрутизації з усіх задіяних інтерфейсів у формі оновлень. RiPv1 виконує широкомовне розсилання цих новлень для 255.255.255.255. Це потребує обробки даних усіма пристроями широкомовної мережі (наприклад, мережа Ethernet). RiPv2 виконує багатоадресну розсилку своїх оновлень на адресу 224.0.0.9. Багатоадресна розсилка потребує меншої пропускної здатності, ніж широкомовні розсилки. Пристрої, не налаштовані для роботи з протоколом RiPv2, відхиляють багатоадресні розсилки на канальному рівні.

Зловмисники часто впроваджують неправильні оновлення, щоб маршрутизатор відправляв дані на помилкову адресу призначення або, щоб значно знизити продуктивність мережі. Неправильні відомості також можуть виявитися в таблиці маршрутизації через помилкове налаштування чи несправну роботу маршрутизатора. Шифрування даних маршрутизації захищає вміст таблиці маршрутизації від маршрутизаторів, що не мають пароля та облікових даних. Протокол RiPv2 має механізм аутентифікації, а RiPv1 – не має.

Незважаючи на безліч удосконалень протоколу RiPv2, він не є зовсім іншим протоколом. Протокол RiPv2 має багато функцій RiPv1, наприклад:

– метрику числа переходів;

– максимальне число переходів, рівне 15;

– TTL, рівне 16 переходам;

– стандартний інтервал відновлення, рівний 30 секундам;

– заборона маршруту, зворотна заборона, поділ горизонту та утримання для уникнення петель;

– оновлення за допомогою UDP-порту 520;

– адміністративна відстань, рівна 120;

– заголовок повідомлення, що вміщує до 25 маршрутів без аутентифікації.

При запуску маршрутизатора кожен інтерфейс, налаштований для роботи з протоколом RIP, відправляє повідомлення-запит. Це повідомлення запитує у всіх сусідів, що працюють по протоколу RIP, відправку повних таблиць маршрутизації. Сусіди, що працюють по протоколу RIP, відправляють повідомлення-відповідь з відомими записами про мережу. Маршрутизатор, отримавши це повідомлення, оцінює кожен маршрут, виходячи з наступних умов:

– якщо запис маршруту новий, маршрутизатор встановлює маршрут у таблиці маршрутизації;

– якщо маршрут уже є в таблиці, а запис надійшов з іншого джерела, існуючий запис буде замінено в таблиці маршрутизації, якщо кількість переходів у новому записі краща;

– якщо маршрут уже є в таблиці і запис надійшов з того ж джерела, існуючий запис буде замінено, навіть якщо метрика не краща.

Запущений маршрутизатор потім відправляє поновлення при включенні з усіх інтерфейсів, що працюють по протоколу RIP і мають свої таблиці маршрутизації. Сусідів, що працюють по протоколу RIP, повідомляють про нові маршрути.

За умови, що маршрутизатори відправляють і обробляють належні версії оновлень маршрутів, протоколи RiPv1 і RiPv2 цілком сумісні. За замовчуванням протокол RiPv2 відправляє та отримує оновлення тільки версії 2. Якщо в мережі необхідно використовувати обидві версії протоколу RIP, адміністратор мережі налаштовує протокол RiPv2 для відправлення та отримання версій 1 і 2. За замовчуванням RiPv1 відправляє відновлення версії 1, а отримує версії 1 і 2.

Налаштування протоколу RiPv2

Перед налаштуванням RiPv2 необхідно призначити IP-адреси та маски всім інтерфейсам, задіяним у маршрутизації. При необхідності потрібно задати тактову частоту для послідовних каналів. Після завершення базового налаштування налаштовується протокол RiPv2.

Базове налаштування RiPv2 складається з трьох команд:

Router(config)#router rip – включення протоколу маршрутизації.

Router(config)#version 2 – визначення версії.

Router(config-router)#network [адреса мережі] – визначення всіх безпосередньо підключених мереж, яким потрібно повідомлення протоколом RIP.

За замовчуванням протокол RiPv2 буде підсумовувати всі мережі, які потрібно оголосити, по своїй класовій границі.

Для оголошень RiPv2 можна задати аутентифікацію.

Протокол RiPv2 поширює маршрут за замовчуванням сусіднім маршрутизаторам разом з оновленнями маршрутів. Для цього потрібно створити маршрут за замовчуванням і додати команду redistribute static у конфігурацію RiPv2

При використанні протоколу RIP може виникнути ряд проблем, пов’язаних із продуктивністю і безпекою. Перша проблема стосується точності таблиці маршрутизації.

Обидві версії протоколу RIP автоматично підсумовують підмережі на межі класу. Це означає, що протокол RIP розпізнає підмережі, як єдину мережу класу A, B чи C. У корпоративних мережах звичайно використовується безкласова IP-адресація і кілька підмереж, деякі з них не зв’язані між собою, у результаті чого утворюються несуміжні підмережі.

На відміну від RiPv1, у протоколі RiPv2 функцію автоматичного підсумовування можна відключити. Якщо функція відключена, RiPv2 буде повідомляти про всі підмережі за рахунок використання мережевої маски. Це необхідно для забезпечення точності таблиці маршрутизації. З цією метою в конфігурацію RiPv2 потрібно додати команду no auto-summary.

Router(config-router)#no auto-summary

Інша проблема – широкомовний режим оновлень RIP. Як тільки конфігурація RIP видає команду network для тієї чи іншої мережі, протокол RIP відразу ж починає відправку повідомлень з усіх інтерфейсів, що входять у цю мережу. Ці оновлення можуть бути потрібні не на всіх ділянках мережі. Наприклад, інтерфейс локальної мережі Ethernet передає ці

оновлення всім пристроям у своєму мережевому сегменті, що створює недоречний трафік. Оновлення маршрутів також може бути перехоплено будь-яким пристроєм, що робить мережу вразливою.

Команда passive-interface, відправлена в режимі інтерфейсу, відключає оновлення маршрутів у визначених інтерфейсах.

Router(config-router)#passive-interface тип_інтерфейсу номер_інтерфейсу

У складних корпоративних мережах, в яких задіяно більше одного протоколу маршрутизації, команда passive-interface визначає маршрутизатори для повідомлення про маршрути RIP. При обмеженні кількості інтерфейсів, що сповіщають про маршрути RIP, підвищується безпека і посилюється контроль над трафіком.

Мережі, яка використовує протокол RIP, потрібно час для конвергенції. Поки не будуть оновлені всі маршрутизатори і в них не буде того ж представлення мережі, деякі з них можуть містити в таблицях маршрутизації недопустимі маршрути.

Помилки у відомостях про мережу можуть викликати у оновленнях маршрутів і трафіку петлі, що ведуть до нескінченної передачі пакету між маршрутизаторами. У протоколі маршрутизації RIP встановлено обмеження на максимальне число переходів – 16.

Механізми уникнення петель маршрутизації протоколу RIP

Петлі маршрутизації негативно позначаються на продуктивності мережі. У протоколі RIP передбачено кілька функцій для усунення цієї проблеми. Ці функції часто поєднуються:

– зворотна заборона (poisoned reverse);

– поділ горизонту (split horizon);

– таймер утримання (holddown timer);

– оновлення при включенні (triggered updates).

Зворотна заборона визначає для метрики маршруту значення 16, і він стає недосяжний. Оскільки протокол RIP визначає нескінченність як 16 переходів, мережа понад 15 переходів недосяжна. Якщо мережа стає неактивною, маршрутизатор змінює метрику для цього маршруту на 16, щоб для всіх інших маршрутизаторів вона була недосяжною. Ця функція запобігає відправленню інформації протоколом маршрутизації по заборонених маршрутах.

Протипетлева функція в протоколі RIP збільшує його стабільність, але збільшує і час конвергенції.

Поділ горизонту запобігає утворенню петель. При передачі декількома маршрутизаторами один одному тих самих маршрутів у мережі можуть утворюватися петлі маршрутизації. Поділ обрію вимагає, щоб маршрутизатор, який отримує інформацію маршрутизації для інтерфейсу, не міг відправити оновлення для тієї ж мережі з цього ж інтерфейсу.

Таймер утримання стабілізує маршрути. Таймер утримання відмовляється приймати оновлення маршрутів з більшою метрикою для тієї ж мережі призначення на період, коли маршрут стає неактивним. Якщо протягом періоду утримання, вихідний маршрут знову стає активним чи маршрутизатор отримує інформацію про маршрут з нижчою метрикою, маршрутизатор встановлює маршрут у таблиці маршрутизації і негайно починає ним користуватися.

Час утримання за замовчуванням дорівнює 180 секундам, у шість разів більше стандартного періоду оновлення. Значення за замовчуванням можна змінити. Однак, будь-який період утримання збільшує час конвергенції і негативно позначається на продуктивності мережі.

Маршрутизація за допомогою протоколу EIGRP

Протокол маршрутизації на основі векторів відстані простий у налаштуванні і вимагає мінімальну кількість ресурсів маршрутизаторів для роботи.

Однак спрощена метрика числа переходів, яка використовується протоколом RIP – це не найточніший спосіб визначення оптимального шляху в складних мережах. Крім того, через обмеження протоколу RIP у 15 переходів, віддалені мережі можуть буи недосяжними.

Протокол RIP виконує періодичні оновлення таблиці маршрутизації, займаючи смугу пропускання, навіть, якщо змін у мережі не було. Маршрутизатори повинні прийняти й обробити ці оновлення, щоб визначити, чи містять вони інформацію про оновлені маршрути.

Переданим від маршрутизатора до маршрутизатора оновленням потрібен час, щоб досягти всіх областей мережі. У результаті маршрутизатори можуть мати у своєму розпорядженні неточне уявлення про мережу. Через великий час конвергенції можуть утворюватися петлі маршрутизації, витрачаючи дорогоцінну пропускну здатність.

Перераховані властивості обмежують застосування протоколу маршрутизації RIP у корпоративному середовищі.

Обмеження RIP протоколу привели до розробки більш удосконалених протоколів з підтримкою VLSM і CIDR, легкою масштабованістю та малим часом конвергенції в складних корпоративних мережах.

Компанія Cisco розробила власний протокол маршрутизації на основі векторів відстані – протокол EIGRP (Enhanced Interior Gateway Routing Protocol). Він наділений розширеними можливостями, що усувають багато обмежень інших протоколів на основі векторів відстані. Крім ряду функцій, спільних з протоколом RIP, протокол EIGRP має багато удосконалених можливостей.

Незважаючи на відносно просте налаштування EIGRP, його функції і параметри носять складний характер. У нього входить безліч функціональних можливостей, що раніше не зустрічалися в жодному іншому протоколі маршрутизації. В силу всіх цих факторів протокол EIGRP – оптимальний вибір для великих багатопротокольних мереж, в яких, в основному, використовуються пристрої компанії Cisco.

Двома основними задачами протоколу EIGRP є забезпечення безпетлевої маршрутизації та швидкої конвергенції. Протокол EIGRP використовує відмінний від протоколу RIP спосіб розрахунку оптимального маршруту. EIGRP використовує складену метрику, яка, головним чином, базується на пропускній здатності та затримці. У порівнянні з числом переходів ця метрика більш точна у визначенні відстані до мережі призначення.

Алгоритм дифузійного відновлення (DUAL), який використовується у протоколі EIGRP, гарантує відсутність петель при розрахунку маршрутів. Коли в топології мережі відбувається зміна, алгоритм DUAL одночасно синхронізує всі пов’язані маршрутизатори. Завдяки цьому, адміністративна відстань для протоколу EIGRP рівна 90, а для протоколу RIP – 120. Менше значення відображає збільшення надійності протоколу EIGRP і підвищення точності метрики. Якщо маршрутизатор отримує маршрути до тієї ж адреси призначення і від RIP, і від EIGRP, він віддасть перевагу маршруту, визначеному протоколом EIGRP.

Протокол EIGRP позначає маршрути, отримані по іншому протоколу маршрутизації, як зовнішні. Оскільки інформація, яка використовується для розрахунку цих маршрутів, менш надійна, ніж метрика EIGRP, маршрутам привласнюється більша адміністративна відстань.

Протокол EIGRP – вдалий вибір для складних корпоративних мереж, в яких, в основному, використовуються маршрутизатори компанії Cisco. Максимальне число переходів протоколу, рівне 255 і дозволяє підтримувати великі мережі. Протокол EIGRP може відображати більше однієї таблиці маршрутизації, оскільки він може збирати і зберігати дані маршрутизації для різних протоколів маршрутизації, наприклад, для IP і IPX. У таблиці маршрутизації EIGRP відображаються маршрути, отримані як всередині, так і зовні локальної системи.

На відміну від інших протоколів на основі векторів відстані протокол EIGRP не відправляє повні таблиці у формі своїх оновлень. EIGRP виконує багатоадресне розсилання часткових оновлень, що стосуються конкретних змін, тільки тим маршрутизаторам, яким ця інформація необхідна, а не всім маршрутизаторам області. Ці оновлення називаються частковими оновленнями, оскільки відображають конкретні параметри.

Замість відправлення періодичних оновлень маршрутів протокол EIGRP відправляє невеликі пакети-вітання для оновлення даних про своїх сусідів. Завдяки малому розміру та частковим оновленням зберігається пропускна здатність і при тому оновлюється інформація про мережу.

Термінологія і таблиці протоколу EIGRP

З метою збереження даних про мережу з оновлень і забезпечення швидкої конвергенції, протокол EIGRP веде ряд таблиць. Маршрутизатори EIGRP розміщують дані про маршрути і топології в ОЗП для забезпечення швидкості відгуку на зміни. Протокол EIGRP веде три взаємозалежні таблиці:

– таблицю сусідів;

– таблицю топології;

– таблицю маршрутизації.

Таблиця сусідів

Таблиця сусідів формує список, що містить дані про безпосередньо підключені сусідні маршрутизатори. EIGRP реєструє адресу виявленого сусіда і підключеного до нього інтерфейсу.

Коли сусід відправляє пакет вітання, він повідомляє про час утримання. Час утримання – проміжок часу, протягом якого маршрутизатор вважає сусіда досяжним. Якщо, протягом часу утримання, пакет-вітання не отриманий, відлік таймера завершується, і алгоритм DUAL виконує перерахунок топології.

Оскільки швидкість конвергенції залежить від точності даних про сусідів, ця таблиця вкрай важлива для роботи протоколу EIGRP.

Таблиця топології

Таблиця топології представляє, у вигляді списку, всі маршрути, отримані від кожного EIGRP-сусіда. Алгоритм DUAL отримує дані з таблиць сусідів і топологій та розраховує найбільш вигідні маршрути до кожної з мереж.

У таблиці топології визначаються до чотирьох основних безпетлевих маршрутів до адреси призначення. Ці кращі маршрути (successor route) відображаються в таблиці маршрутизації. Протокол EIGRP може розподіляти навантаження, тобто відправляти пакети за адресою призначення з допомогою декількох шляхів. Розподіл навантаження виконується за допомогою кращих маршрутів, що одночасно можуть бути з рівною вартістю і з нерівною вартістю. Ця функція дозволяє уникнути перевантаження пакетами того чи іншого маршруту.

Резервні маршрути, названі можливими спадкоємцями (feasible successor), відображаються в таблиці топології, але відсутні в таблиці маршрутизації. Якщо не діє основний маршрут, кращим маршрутом стає можливий спадкоємець. Це заміщення відбувається за умови, що оголошена відстань feasible successor менша допустимої відстані поточного successor до адреси призначення.

Таблиця маршрутизації

Якщо в таблиці топології розміщені дані про безліч різних маршрутів до мережі призначення, то в таблиці маршрутизації відображаються тільки оптимальні маршрути, які називаються кращими маршрутами (successor route).

Протокол EIGRP відображає інформацію про маршрути двома способами:

– у таблиці маршрути, отримані по протоколу EIGRP, позначаються символом D;

– EIGRP позначає динамічні і статичні протоколи, отримані від інших протоколів або не з мережі EIGRP, як D EX, або зовнішні, оскільки вони надійшли не від маршрутизаторів EIGRP цієї ж автономної системи.

Сусіди і суміжники EIGRP

Щоб протокол EIGRP зміг обмінюватись пакетами між маршрутизаторами, йому необхідно спочатку знайти своїх сусідів. Сусіди EIGRP – це інші маршрутизатори, що працюють по протоколу EIGRP у безпосередньо підключених мережах із загальним доступом.

Маршрутизатори EIGRP використовують пакети вітань для виявлення сусідів і встановлення примикань із сусідніми маршрутизаторами. За замовчуванням у каналах зі швидкістю більше T1 відбувається багатоадресна розсилка пакетів-вітань через кожні 5 секунд, а в каналах зі швидкістю T1 і менше – через кожні 60 секунд.

В IP-мережах адресою багатоадресної розсилки є 224.0.0.10. У пакет-вітання входять: інформація про інтерфейси маршрутизаторів і адреси інтерфейсів. Маршрутизатор EIGRP вважає, що поки надходять пакети-вітання від сусіда, сусід і його маршрути досяжні.

Час утримання – це період очікування протоколом EIGRP пакету-вітання. Звичайний час утримання в три рази більший від інтервалу вітання. Після закінчення часу утримання, алгоритм DUAL виконує перерахунок топології та оновлює таблицю маршрутизації.

Дані, виявлені за допомогою протоколу вітання, надходять у таблицю сусідів. У рядку з порядковим номером записується кількість останніх отриманих вітань від кожного сусіда і встановлюється мітка часу в момент надходження пакету-вітання.

Після встановлення сусідства протокол EIGRP використовує пакети різних типів для обміну і відновлення даних у таблицях маршрутизації. Сусіди отримують повідомлення про нові, недосяжні та знову виявлені маршрути шляхом обміну цими пакетами:

– підтвердження;

– оновлення;

– запит;

– відповідь.

При втраті маршруту він переходить в активний стан, а алгоритм DUAL виконує пошук нового маршруту до адреси призначення. Після виявлення маршруту він розміщується в таблиці маршрутизації і переходить у пасивний стан.

За допомогою цих пакетів алгоритм DUAL збирає дані, необхідні для розрахунку оптимального маршруту до мережі призначення.

Пакет підтвердження означає отримання пакету-оновлення або запиту відповіді. Пакети підтверджень – це невеликі пакети-вітання без даних. Ці типи пакетів завжди одноадресні.

Пакет-оновлення відправляє дані про топологію мережі своєму сусіду. Цей сусід оновлює свою таблицю топології. Для відправлення всіх даних про топологію новому сусіду іноді потрібно відправити кілька пакетів оновлення.

Кожного разу, коли алгоритм DUAL переводить маршрут в активний стан, маршрутизатор має відправити пакет запиту всім сусідам. Сусіди у свою чергу повинні відправити відповідь, навіть, якщо в ній буде зазначено, що про адресу призначення інформації немає. Дані кожного пакету-відповіді дозволяють алгоритму DUAL знайти кращий маршрут до мережі призначення. Запити можуть бути багатоадресними та одноадресними. Відповіді завжди одноадресні.

Пакети EIGRP використовують або TCP, або UDP протокол. Пакети оновлення, запиту і відповіді використовують службу типу TCP, а підтвердження і пакети-вітання – службу типу UDP.

Будучи протоколом маршрутизації, EIGRP функціонує незалежно від мережевого рівня. Компанія Cisco розробила власний протокол четвертого рівня – надійний транспортний протокол (RTP – Reliable Transport Protocol). RTP гарантує доставку та отримання пакетів EIGRP для всіх протоколів мережевого рівня. Оскільки у великих складних мережах може використовуватися ряд протоколів мережевого рівня, цей протокол забезпечує гнучкість і масштабованість EIGRP.

RTP можна використовувати одночасно і як транспортний протокол з гарантованою доставкою, і як транспортний протокол з негарантованою доставкою, подібно TCP і UDP. При RTP з гарантованою доставкою отримувач повинен відправити відправнику пакет підтвердження. Пакети оновлення, запиту і відповіді відправляються в режимі гарантованої доставки, а пакети-вітання і підтвердження – у режимі негарантованої доставки і не вимагають підтвердження. RTP використовує як одноадресні, так і багатоадресні пакети. Багатоадресні пакети EIGRP використовують зарезервовану адресу багатоадресної розсилки 224.0.0.10.

Кожен протокол мережевого рівня працює через протокол-залежний модуль (PDM – Protocol Dependent Module), відповідальний за конкретну задачу маршрутизації. Усі модулі PDM ведуть три таблиці. Наприклад, у маршрутизатора, на якому запущені IP, IPX і AppleTalk, є три таблиці сусідів, три таблиці топології і три таблиці маршрутизації.

Метрики і конвергенція протоколу EIGRP

Для визначення кращого маршруту до адреси призначення EIGRP використовує складене значення метрики. Ця метрика визначається на основі наступних значень:

– смуга пропускання;

– затримка;

– надійність;

– навантаження.

Ще одне значення – максимальний розмір переданого блоку даних (MTU) – входить у оновлення маршрутів, але не є метрикою маршрутизації.

У формулу складеної метрики входять коефіцієнти K: з K1 до K5.

За замовчуванням для K1 і K3 установлюється значення 1, а для K2, K4 і K5 – 0. Коефіцієнт 1 означає, що пропускна здатність і затримка мають однакову вагу при розрахунку складеної метрики.

Пропускна здатність

Метрика пропускної здатності є статичним значенням, відображається у Кбіт/с. У більшості серійних інтерфейсів значення пропускної здатності за замовчуванням дорівнює 1544 Кбіт/с. Ця метрика відображає пропускну здатність підключення T1.

Іноді значення пропускної здатності може не відображати фактичну фізичну пропускну здатність інтерфейсу. Пропускна здатність впливає на розрахунок метрики і, як наслідок, на вибір шляху EIGRP. Якщо в з'єднання з пропускною здатністю 56 Кбіт/с надходить повідомлення зі значенням 1544 Кбіт/с, воно може негативно позначитися на конвергенції, оскільки необхідно буде перебороти навантаження трафіку.

Іншими метриками, що використовує EIGRP для розрахунку вартості каналу, є затримка, надійність і навантаження.

Метрика затримки – статичне значення на основі типу вихідного інтерфейсу. Значення за замовчуванням дорівнює 20 000 мікросекунд для серійних інтерфейсів і 100 мікросекунд для інтерфейсів Fast Ethernet.

Метрика затримки не відображає фактичну кількість часу, що затрачають пакети, щоб досягти адреси призначення. При зміні значення затримки, пов'язаного з визначеним інтерфейсом, змінюється метрика, але це не має фізичного впливу на мережу.

Метрика надійності означає частоту помилок у каналі. На відміну від затримки метрика надійності оновлюється автоматично в залежності від умов каналу. Її значення дорівнює від 0 до 255. Надійність, рівна 255/255, показує канал зі стовідсотковою надійністю.

Навантаження відображає об'єм трафіку в каналі. Менше значення навантаження переважає високе. Наприклад, значення 1/255 означає канал з мінімальним навантаженням, а 255/255 – канал, завантажений на 100%.

У таблиці топології EIGRP метрики використовуються для розрахунку значень можливої відстані (FD) і оголошеної (AD) або заявленої відстані (RD). Алгоритму DUAL ці значення необхідні для визначення кращих шляхів і можливих спадкоємців (successors and feasible successors).

Можлива відстань (Feasible distance) – це краща метрика EIGRP по шляху до адреси призначення від маршрутизатора.

Оголошена відстань (Advertised distance) – це краща метрика, отримана від сусіднього маршрутизатора.

Безпетлевий маршрут з найменшою можливою відстанню стає кращим маршрутом (successor route). Можлива наявність декількох кращих маршрутів до адреси призначення в залежності від фактичної топології. Можливим спадкоємцем (feasible successor) є маршрут, оголошена відстань якого менша можливої відстані кращого маршруту.

Алгоритм DUAL виконує конвергенцію відразу ж після зміни топології. Алгоритм DUAL зберігає можливих спадкоємців у таблиці топології і відправляє в таблицю маршрутизації кращого з них, як кращий маршрут. При відсутності можливих спадкоємців вихідний маршрут переходить в активний режим, і відправляються запити на пошук нового спадкоємця.

Впровадження протоколу EIGRP

Налаштування протоколу EIGRP

Налаштування базового EIGRP відносно просте. Його налаштування багато в чому повторює протокол RiPv2.

Щоб почати процес маршрутизації EIGRP, використовуються два кроки.

Крок 1

Включити процес маршрутизації EIGRP.

Для включення процесу EIGRP необхідний параметр автономної системи. Цьому параметру AS можна призначити будь-яке 16-розрядне значення, і він визначає всі маршрутизатори однієї організації. Незважаючи на те, що EIGRP розглядає параметр, як номер автономної системи, він фактично виступає в ролі ідентифікатора процесу. Цей номер AS має тільки локальне значення і відрізняється від номера автономної системи, який видається і контролюється Комітетом з цифрових адрес в Інтернет (IANA –Internet Assigned Numbers Authority).

Номер AS у команді повинен збігатися для всіх маршрутизаторів, що беруть участь у процесі маршрутизації EIGRP.

Крок 2

Оголосити мережі, про які потрібно передати інформацію.

Команда network вказує протоколу EIGRP, які мережі та інтерфейси беруть участь у процесі EIGRP.

Щоб налаштувати EIGRP для передачі інформації лише про деякі підмережі, потрібно додати шаблонну маску (wildcard mask) після номеру мережі. Щоб визначити шаблонну маску, потрібно відняти мережеву маску з 255.255.255.255.

У деяких версіях Cisco IOS замість шаблонної маски можна вказати маску підмережі. Навіть при використанні звичайної маски, шаблонна маска буде відображена у вихідних даних команди show running-config.

Стандартну базову конфігурацію EIGRP завершують дві додаткові команди.

Команда eigrp log-neighbor-changes додається для перегляду змін сусідів. Ця функція дозволяє адміністраторам відслідковувати стабільність мережі EIGRP.

Для послідовних каналів, що не відповідають пропускній здатності EIGRP у 1,544 Мбіт/с, варто додати команду bandwidth з вказанням фактичної швидкості каналу (у Кбіт/с). Неточне задання пропускної здатності ускладнює вибір оптимального маршруту.

Після включення EIGRP усі маршрутизатори, налаштовані для роботи з EIGRP та заданим правильним номером автономної системи, працюють по EIGRP. Це означає, що маршрутизатори з іншою інформацією про маршрутизацію можуть негативно впливати і навіть зашкодити таблицям маршрутизації. Щоб цього уникнути, можна включити аутентифікацію в конфігурації EIGRP. Після налаштування аутентифікації маршрутизатор перевіряє достовірність джерела оновлень маршрутів перед тим, як їх прийняти.

Для аутентифікації EIGRP потрібно мати ключ. Протокол EIGRP дозволяє адміністраторам керувати ключами через ланцюжок ключів. Налаштування аутентифікації EIGRP складається з двох кроків: створення ключа і включення аутентифікації з його використанням.

Створення ключа

Щоб створити ключ, потрібно виконати такі команди.

key chain ім’я ланцюжка

– команда глобального налаштування.

– вказується ім’я ланцюжка ключів і введення для неї режиму налаштування.

key ідентифікатор ключа

– визначення номеру ключа і введення режиму налаштування для цього ідентифікатору ключа.

key-string текст

– вказання рядка ключа або паролю. Це налаштування має збігатися у всіх маршрутизаторів EIGRP.

Включення аутентифікації

Ключ служить для включення аутентифікації MD5 для EIGRP за допомогою таких команд налаштування інтерфейсу:

ip authentication mode eigrp md5

– вказує на необхідність аутентифікації MD5 для обміну пакетами EIGRP.

ip authentication key-chain eigrp ім'я_ланцюжка AS

– AS позначає автономну систему конфігурації EIGPR.

Параметр імені ланцюжка вказує ланцюжок ключів, які були налаштовані раніше.

Підсумовування маршрутів EIGRP

Як і протокол RIP, EIGRP автоматично підсумовує на межі класу мережі з наявністю підмереж. Для сумарного маршруту EIGRP створює тільки один запис у таблиці маршрутизації. З сумарним маршрутом пов'язується оптимальний маршрут, або кращий маршрут (successor route). В результаті весь трафік, призначений для підмереж, надходить по цьому єдиному шляху.

У корпоративній мережі обраний шлях до сумарного маршруту може бути не оптимальним для трафіку, призначеного для тієї чи іншої окремої підмережі. Маршрутизатори можуть знайти оптимальні маршрути до кожної з окремих підмереж, тільки отримавши від сусідів інформацію про ці підмережі.

Якщо відключене підсумовування за замовчуванням, то оновлення будуть містити інформацію про підмережі. У таблиці маршрутизації будуть встановлені записи для кожної підмережі, а також запис для сумарного маршруту. Сумарний маршрут називається батьківським маршрутом, а маршрути підмереж – дочірніми маршрутами.

Для всіх батьківських маршрутів у таблиці маршрутизації EIGRP встановлює сумарний маршрут Null0. Інтерфейс Null0 означає, що це не фактичний маршрут, а деяке підсумовування з метою передачі даних про маршрутизацію. Якщо пакет відповідає одному з дочірніх маршрутів, він направляється з відповідного інтерфейсу. Якщо пакет відповідає сумарному маршруту і не відповідає жодному з дочірніх, він буде відхилений.

При використанні підсумовування за замовчуванням розміри таблиць маршрутизації скорочуються. Якщо підсумовування відключити, то розміри оновлень і таблиць будуть збільшуватися. Необхідність автоматичного підсумовування визначається, виходячи із загальної продуктивності мережі і моделей трафіку.

Для відключення підсумовування за замовчуванням, використовується команда no auto-summary.

Якщо автоматичне підсумовування відключити, то буде надходити інформація про всі підмережі. Адміністратор може зіштовхнутися із ситуацією, коли одним підмережам підсумовування необхідне, а іншим - непотрібне. Рішення про підсумовування приймається, виходячи з розташування підмереж. Наприклад, ізольовані підмережі, підключені до одного маршрутизатора, підходять для підсумовування.

Підсумовування вручну покращує керування маршрутами EIGRP. За допомогою цієї функції адміністратор визначає, які підмережі і на яких інтерфейсах повідомляти сумарними маршрутами.

Підсумовування вручну виконується на рівні інтерфейсу і наділяє адміністратора всією повнотою керування. Підсумований вручну маршрут відображається в таблиці маршрутизації у вигляді маршруту EIGRP на основі логічного (не фізичного) інтерфейсу:

D 192.168.0.0/22 is a summary, Null0

Перевірка роботи протоколу EIGRP

Незважаючи на відносну простоту налаштування протоколу EIGRP, він використовує складні технології, щоб перебороти обмежені можливості протоколів маршрутизації на основі векторів відстані. Щоб правильно перевірити, знайти та усунути помилки в конфігурації мережі, що працює по протоколу EIGRP, важливо розуміти ці технології. У число доступних команд перевірки входять такі команди.

show ip protocols

– Перевіряє, чи оголошені протоколом EIGRP відповідні мережі.

– Відображає номер автономної системи та адміністративну відстань.

show ip route

– Перевіряє наявність отриманих маршрутів EIGRP у таблиці маршрутизації.

– Позначає маршрути EIGRP символами D чи D EX.

– Відображає для внутрішніх маршрутів адміністративну відстань за замовчуванням, рівну 90.

show ip eigrp neighbors detail

– Перевіряє суміжні EIGRP форми.

– Відображає IP-адреси та інтерфейси сусідніх маршрутизаторів.

show ip route

– Відображає кращі маршрути і всіх можливих спадкоємців (successors and all feasible successors).

– Відображає можливу та оголошену відстань (feasible distance and reported distance).

show ip eigrp interfaces detail

– Перевіряє інтерфейси, що використовують EIGRP.

show ip eigrp traffic

– Відображає кількість і типи EIGRP пакетів, що відправляються та отримуються.

Однієї з основних задач цих команд show є перевірка правильності утворення EIGRP adjacencies і обміну пакетами EIGRP між маршрутизаторами. Робота EIGRP неможлива, якщо не сформовано adjacencies, тому їх потрібно перевірити до пошуку й усунення інших помилок.

Якщо з adjacencies все в порядку, а проблеми як і раніше залишаються, адміністратору варто почати пошук і усунення помилок за допомогою команд налагодження для перегляду інформації про роботу EIGRP на маршрутизаторі в режимі реального часу.

debug eigrp packet

– відображає передачу й отримання всіх пакетів EIGRP.

debug eigrp fsm

– відображає активність можливого спадкоємця, щоб визначити стан маршрутів (виявлені, установлені чи видалені протоколом EIGRP).

Операції налагодження вимагають значної пропускної здатності та обчислювальних потужностей маршрутизатора, особливо налагодження дуже складних протоколів типу EIGRP. Ці команди дозволяють визначити джерело втрати маршруту або відсутньої суміжності EIGRP, але продуктивність мережі при використанні таких команд може падати.

Проблеми й обмеження протоколу EIGRP

Хоча протокол маршрутизації EIGRP досить функціональний, проте , має деякі обмеження:

– не працює в середовищах різних провайдерів, оскільки є власним протоколом компанії Cisco;

– найоптимальніше функціонує в мережах плоского типу;

– у маршрутизаторів повинна збігатися автономна система, і його неможливо розділити на групи;

– може створювати дуже великі таблиці маршрутизації, що вимагають великих пакетів оновлень і пропускної здатності;

– використовує більший об’єм пам’яті та обчислювальних ресурсів у порівнянні з протоколом RIP;

– працює ефективно, якщо не змінювати параметри за замовчуванням;

– для його обслуговування необхідні адміністратори з поглибленими технічними знаннями.

Протокол EIGRP забезпечує оптимальну маршрутизацію на основі векторів відстані, використовуючи додаткові функції, звичайно характерні для протоколів маршрутизації на основі стану каналів, у тому числі, обмежені оновлення і суміжності сусідів. Для успішного впровадження багатьох функцій протоколу EIGRP потрібно ретельне налаштування, моніторинг, пошук і усунення помилок.

Основним питанням безпеки корпоративних мереж є баланс між двома важливими вимогами: необхідністю відкрити мережі для розвитку бізнес-можливостей та необхідністю захисту приватної, особистої та стратегічної бізнес-інформації.

Застосування ефективної політики безпеки є найбільш важливим кроком, який організація може зробити для захисту своєї мережі. Вона містить рекомендації про заходи, які будуть проводитися, та ресурси, які будуть використовуватися для захисту корпоративної мережі.

Значення та розміри комп'ютерних мереж швидко зростають в часі. Якщо не забезпечується безпека мережі, то це може мати дуже серйозні наслідки, такі як втрата конфіденційності, крадіжка інформації, несанкціонований доступ до ресурсів. Типи потенційних загроз для безпеки мережі постійно розвиваються, що ускладнює питання безпеки.

Розвиток електронного бізнесу, мобільної комерції та безпровідних мереж вимагає інтегрованих та гнучких рішенн безпеки, оскільки інструменти та методи мережевих атак також швидко розвиваються. Наприклад, в 1985 році зловмисник повинен був мати складний комп'ютер, мати знання програмування і мереж, щоб використовувати елементарні засоби та основні атаки. Проте, з часом методи та інструменти атак вдосконалювались, і сьогодні зловмисникам не потрібен такий же рівень складності знаннь. Люди, які раніше не брали участі в комп'ютерних злочинах, в даний час взмозі це зробити

Маршрутизація на основі стану каналу

На відміну від протоколів вектора відстані протоколи маршрутизації типу стану каналу можуть створити "повну картину", або топологію мережі шляхом збору інформації від усіх інших маршрутизаторів.

Протоколи стану каналу використовують інформацію про стан каналу для створення топологічної карти та вибору найкращого шляху до всіх мереж призначення в топології.

Протоколи стану каналу не використовують періодичні оновлення. Після того, як наступила конвергенція мережі, оновлення надсилається тільки тоді, коли відбувається зміна в топології.

Корпоративні мережі і провайдери використовують протоколи на базі стану каналу, що пов'язано з їх ієрархічною структурою та можливістю масштабування для великих мереж. Протоколи маршрутизації на основі векторів відстаней не є правильним вибором для складної корпоративної мережі.

Протоколи маршрутизації на основі стану каналу, також відомі, як протоколи найкоротшого шляху побудовані на використанні алгоритму Дейкстри (Edsger Dijkstra's shortest path first (SPF) algorit). До них належать:

– Open Shortest Path First (OSPF);

– Intermediate System-to-Intermediate System (IS-IS).

Алгоритм Дейкстри, або алгоритм коротшого шляху враховує вартість кожної ділянки шляху від відправника до отримувача.

Кожен маршрутизатор визначає вартість до кожного пункту призначення. Іншими словами, кожен маршрутизатор обчислює алгоритм SPF та визначає вартість із своєї власної точки зору.

Принцип роботи протоколів стану каналу такий:

1. Кожен маршрутизатор дізнається про свої безпосередньо підключені мережі. Це здійснюється завдяки виявленню, що інтерфейс знаходиться в активному стані.

2. Кожен маршрутизатор, несе відповідальність за роботу зі своїми сусідами з безпосередньо під'єднаних мереж. Як і в EIGRP, маршрутизатори стану каналу обмінюються пакетами-вітаннями з іншими маршрутизаторами, які підтримують протокол стану каналу.

3. Кожен маршрутизатор створює LSP пакет (Link-State Packet), що містить стан кожного безпосередньо під'єднаного каналу. Це здійснюється шляхом запису відповідної інформації про кожного сусіда, у тому числі його ідентифікатор (neighbor ID), тип каналу та пропускну здатність.

4. Кожен маршрутизатор розсилає пакети LSP до всіх сусідів, які зберігають всі отримані LSP в базі даних. Сусіди потім передають LSP пакети своїм сусідам. Таким чином

всі маршрутизатори в домені протоколу отримають LSP пакети. Кожен маршрутизатор зберігає копію кожного LSP, отриманого від своїх сусідів у локальній базі даних.

5. Кожен маршрутизатор використовує базу даних, щоб побудувати повну карту топології і обчислити оптимальний шлях до кожної мережі призначення. Алгоритм SPF використовується для побудови карти топології та визначення найкращого шляху до кожної з мереж.

Канал (Link)

В протоколах маршрутизації стану каналу, link – це інтерфейс маршрутизатора. Як і в протоколах вектора відстані та статичній маршрутизації, інтерфейс повинен бути налаштований з правильною IP-адресою та мережевою маскою і повинен бути в піднятому стані, щоб протокол стану каналу міг дізнатися про link.

Стан каналу (Link-State)

Це інформація про стан каналу (link-states). Ця інформація включає в себе:

– IP-адресу інтерфейсу та маску підмережі.

– Тип мережі (Ethernet, Serial point-to-point link).

– Вартість каналу (cost of link).

– Сусідні (neighbor) маршрутизатори.

Протоколи маршрутизації на базі стану каналу мають ряд переваг в порівнянні з протоколами вектора відстані.

Будують топологічну карту.

Протоколи стану каналу створюють топологічну карту, або SPF дерево топології мережі. Протоколи вектора відстані не мають топологічної карти мережі. Маршрутизатори з підтримкою протоколу вектора відстані мають тільки список мереж, який включає в себе вартість (відстань) та маршрутизатори наступного переходу (напрям) до цих мереж. Оскільки протоколи стану каналу обмінюються повідомленнями про стан каналу, алгоритм SPF може побудувати SPF дерево мережі. Використовуючи дерево SPF, кожен маршрутизатор може самостійно визначити найкоротший шлях до будь-якої мережі.

Швидка конвергенція.

Отримавши LSP пакет протоколи стану каналу відразу ж відправляють LSP пакет через всі інтерфейси, крім інтерфейсу, з якого був отриманий LSP пакет. Протоколи вектора відстані повинні обробити кожне оновлення маршрутизації та оновити таблицю маршрутизації перед надсиланням на інші інтерфейси.

Оновлення по події.

Після початкового надсилання LSP, протоколи стану каналу відправляють LSP лише тоді, коли є зміни в топології. LSP містить інформацію лише про змінені стани каналів. Протоколи стану каналу не надсилають періодичних оновлень.

Ієрархічний дизайн.

Протоколи маршрутизації стану каналу використовують концепцію областей, що дозволяє створювати ієрархічний дизайн мереж для агрегації (сумування) маршрутів та ізолювати проблеми маршрутизації в межах однієї області.

Маршрутизація з використанням протоколу OSPF

Принцип роботи протоколу

Протокол OSPF (Open Shortest Path First) – приклад протоколу маршрутизації на базі стану каналу. Протокол OSPF – це відкритий стандарт протоколу маршрутизації, розроблений інженерною групою по розвитку Інтернет (IETF) для підтримки IP-трафіку.

Протокол OSPF є безкласовим протоколом внутрішніх шлюзів (IGP). Він поділяє мережу на різні секції, що називають областями. Даний поділ дає можливість більшого масштабування. Робота з декількома областями дозволяє адміністратору мережі вибірково включати підсумовування маршрутів та ізолювати проблеми з маршрутизацією в межах якої-небудь однієї області.

Протоколи маршрутизації на базі стану каналу, такі як OSPF, не надсилають періодичних розсилань повної таблиці маршрутизації. Замість цього після конвергенції мережі відправлення оновлення протоколом на базі стану каналу здійснюється тільки при

якій-небудь зміні в топології мережі, наприклад, при відключенні каналу. Крім цього, кожні 30 хвилин протокол OSPF виконує повне оновлення.

Протоколи маршрутизації на базі стану каналу, такі як OSPF, працюють нормально у великих ієрархічних мережах, де важлива швидка збіжність.

У порівнянні з протоколами векторів відстані, для протоколів маршрутизації по стану каналу потрібно:

– більш складний процес планування і конфігурації мережі;

– збільшені ресурси маршрутизатора;

– більший об'єм пам'яті для збереження великої кількості таблиць;

– більш висока потужність процесора та обчислювальна потужність для складних розрахунків маршрутизації.

Однак при високій обчислювальній потужності маршрутизаторів, доступній в даний час, виконання цих вимог не є складним.

Маршрутизатори, на яких виконуються протоколи RIP, отримують оновлення від маршрутизаторів, що знаходяться в безпосередньому сусідстві, але без докладної інформації про всю мережу. Маршрутизатори, на яких виконуються протоколи OSPF, створюють повну карту мережі зі своєї точки зору. Дана карта дозволяє їм швидко визначати безпетлеві альтернативні маршрути у випадку відмови якого-небудь мережевого каналу.

Протокол OSPF використовує 5 типів LSP пакетів, кожен з яких має певне призначення в процесі роботи маршрутизації OSPF [12]:

1. Hello – Hello пакети використовуються для встановлення та керування суміжними відносинами з іншими OSPF маршрутизаторами.

2. DBD (The Database Description) – пакети містять скорочені списки бази даних стану лінків та використовуються при отриманні машрутизатором для звірки з локальною базою даних.

3. LSR (Link-State Request) – маршрутизатори можуть запитувати додаткову інформацію про записи в DBD шляхом надсилання запиту LSR.

4. LSU (Link-State Update) – пакети використовуються для відповіді на LSR запити, а також для оголошення нової інформації. LSU пакети містять сім типів оголошень LSA (Link-State Advertisements).

5. LSAck (Link-State Acknowledgement) – коли маршрутизатор отримує LSU він надсилає підтвердження про отримання LSAck.

Протокол OSPF не виконує автоматичного підсумовування на границях головної мережі. Крім того, у рішеннях по протоколах OSPF, пропонованих компанією Cisco, для визначення вартості каналу використовується пропускна здатність. Ця метрика вартості використовується протоколом OSPF для визначення найкращого маршруту. Канал з більш високою пропускною здатністю забезпечує нижчу вартість.

Для встановлення найкоротшого шляху маршрутизатор більше довіряє метриці, оснований на пропускній здатності, ніж метриці, оснований на числі переходів. Адміністративна відстань протоколу OSPF дорівнює 110, що менше, ніж для RIP, завдяки точності метрики.

Метрики і конвергенція протоколу OSPF

Протокол OSPF використовує метрику вартості для окремого каналу на основі його пропускної здатності або швидкості. Метрикою для конкретної мережі призначення є сума вартості всіх каналів шляху. Якщо існує кілька шляхів до мережі, кращим є шлях з найменшою вартістю, і він заноситься в таблицю маршрутизації.

Для розрахунку вартості каналу протоколу OSPF використовується рівняння:

вартість = 100 000 000 / пропускна здатність каналу в біт/с.

Значення пропускної здатності для рівняння дає пропускна здатність, що налаштовується в інтерфейсі. Пропускна здатність інтерфейсу визначається командою show interfaces.

При швидкостях каналу 100 Мбіт/с і вище, наприклад, як у каналів Fast Ethernet і Gigabit Ethernet, використання даного рівняння викликає труднощі. Незалежно від різниці у швидкості між цими двома каналами, вони обоє розраховуються до значення 1, тому,

незважаючи на розходження цих каналів, вони будуть оброблятися однаково. Щоб це компенсувати, варто налаштувати значення вартості інтерфейсу вручну за допомогою команди ip ospf cost.

У межах однієї області маршрутизатори OSPF повідомляють інформацію про стан своїх з'єднань сусіднім маршрутизаторам. Для оголошення інформації про стан каналів використовуються повідомлення, що називаються оголошеннями про стан каналу (LSA).

Після отримання оголошень LSA з описом усіх каналів у межах відповідної області маршрутизатор OSPF використовує алгоритм SPF (алгоритм Дейкстри) для створення топологічної деревоподібної схеми, або карти мережі. Кожен маршрутизатор, на якому виконується даний алгоритм, визначає себе як кореневий елемент свого власного дерева SPF. Починаючи від кореневого елемента, дерево SPF визначає найкоротший шлях до кожного місця призначення і загальну вартість кожного шляху.

Інформація про дерево SPF зберігається в базі даних топології. Маршрутизатор заносить найкоротший шлях до кожної мережі в таблицю маршрутизації.

Конвергенція досягається, якщо всі маршрутизатори:

– отримають інформацію про кожне місце призначення в мережі;

– оброблять дану інформацію з використанням алгоритму SPF;

– обновлять свої таблиці маршрутизації.

При використанні протоколів OSPF оновлення інформації про стан каналів розсилаються з появою в мережі яких-небудь змін. Але яким чином маршрутизатор може довідатися про відмову сусіднього маршрутизатора? Маршрутизатори OSPF встановлюють і підтримують сусідські відносини, чи відносини суміжності, з іншими маршрутизаторами OSPF, підключеними до мережі. Суміжність – це покращена форма сусідських відносин між маршрутизаторами, що бажають обмінюватися інформацією про маршрутизацію. При ініціації маршрутизаторами відносин суміжності із сусідніми маршрутизаторами починається обмін оновленнями інформації про стан каналів. Маршрутизатори досягають стану суміжності FULL (повний), коли вони мають синхронізовані дані у своїй базі даних станів каналів.

Перед тим, як стати повністю суміжним із сусіднім маршрутизатором, той чи інший маршрутизатор проходить через кілька змін стану.

– Init (ініціація);

– 2-Way (двосторонній режим);

– Exstart;

– Exchange (обмін інформацією);

– Loading (завантаження);

– Full (повний).

У OSPF протокол-вітання використовується для початкового встановлення і ведення відносин суміжності. Протокол-вітання посилає дуже маленькі пакети-вітання до підключеного маршрутизатора OSPF на адресу багатоадресної розсилки 224.0.0.5. Пакети надсилаються кожні 10 секунд по каналах Ethernet і широкомовних каналах, та кожні 30 секунд по не широкомовних каналах. Пакети-вітання також містять у собі налаштування маршрутизатора. Налаштування містять інтервал вітання, паузу, тип мережі, а також тип аутентифікації і дані аутентифікації, якщо вона налаштована. Для встановлення суміжності між будь-якими двома маршрутизаторами всі налаштування повинні збігатися. Маршрутизатор записує виявлені відносини суміжності між сусідніми маршрутизаторами в базу даних відносин суміжності.

Сусідні маршрутизатори OSPF та відносини суміжності

Стан Full є стандартним для маршрутизатора OSPF. Якщо маршрутизатор тривалий час знаходиться в іншому стані, це вказує на наявність якої-небудь проблеми, наприклад, розбіжності налаштувань. Єдиним виключенням є стан 2-way. У широкомовному оточенні маршрутизатор досягне стану full тільки з призначеним маршрутизатором (DR) і резервним призначеним маршрутизатором (BDR). Всі інші сусідні маршрутизатори будуть відображатися в стані 2-way.

Задачею маршрутизаторів DR і BDR є зниження кількості посилань оновлених даних, непотрібного трафіку і непродуктивних процесів на всіх маршрутизаторах. Це досягається шляхом відправлення на всі маршрутизатори запиту про прийом обновлених даних тільки від маршрутизатора DR. У широкомовних сегментах мережі є тільки по одному маршрутизатору DR і BDR. Всі інші маршрутизатори повинні мати з'єднання з маршрутизатором DR і BDR. При відмові якого-небудь каналу маршрутизатор, що має інформацію про даний канал, посилає інформацію на маршрутизатор DR, використовуючи адресу багатоадресної розсилки 224.0.0.6. Маршрутизатор DR відповідає за розсилання інформації про зміну на всі інші маршрутизатори OSPF за адресою багатоадресної розсилки 224.0.0.5. Крім зниження кількості оновлень, що розсилаються по мережі, цей процес також забезпечує отримання всіма маршрутизаторами однакової інформації в той самий час і з одного джерела.

Маршрутизатор BDR забезпечує відсутність яких-небудь критичних точок. Як і маршрутизатор DR, маршрутизатор BDR спостерігає за адресою 224.0.0.6 і отримує усі оновлення, що посилаються на маршрутизатор DR. При відмові маршрутизатора DR маршрутизатор BDR отримує функції маршрутизатора DR, і призначається новий маршрутизатор BDR. Будь-який маршрутизатор, не обраний як маршрутизатор DR чи BDR, називається маршрутизатором DROther.

Маршрутизатором DR призначається маршрутизатор з найвищим, у межах локальної мережі, ідентифікатором маршрутизатора. Маршрутизатором BDR призначається маршрутизатор з другим по величині ідентифікатором.

Ідентифікатором маршрутизатора є IP-адреса, що визначається таким чином:

1. Значенням, налаштованим з використанням команди router-id.

2. Якщо за допомогою команди router-id не встановлено ніякого значення, то вищою IP-адресою, налаштованою на loopback-інтерфейсі.

3. Якщо відсутній який-небудь налаштований loopback-інтерфейс, то вищою IP-адресою у будь-якому активному фізичному інтерфейсі.

Ідентифікатор маршрутизатора можна переглядати за допомогою наступних show команд:

show ip protocols, show ip ospf та show ip ospf interface.

У деяких випадках адміністратору може знадобитися можливість призначити в ролі маршрутизаторів DR і BDR які-небудь конкретні маршрутизатори. Це можуть бути маршрутизатори з більшою обчислювальною потужністю або з меншим завантаженням трафіку. Адміністратор може примусово призначити маршрутизатори DR і BDR шляхом налаштування пріоритету з використанням команди налаштування інтерфейсу:

ip ospf priority номер

За замовчуванням маршрутизатори OSPF мають значення пріоритету 1. При зміні значення пріоритету на якому-небудь маршрутизаторі, в ролі маршрутизатора DR буде обраний маршрутизатор з вищим налаштованим значенням пріоритету незалежно від вищого ідентифікатора маршрутизатора. Найвищим значенням для налаштування пріоритету маршрутизатора є значення 255. Значення 0 означає, що маршрутизатор не можна вибрати як маршрутизатор DR чи BDR.

Не для всіх типів каналів потрібно маршрутизатор DR чи BDR. Типи каналів, обумовлені протоколом OSPF, містять у собі:

Мережі із широкомовним розсиланням.

– Ethernet.

– Мережі «точка-точка» (PPP).

– послідовні;

– T1/E1.

Нешироковомовні мережі множинного доступу (NBMA).

– Frame Relay;

– ATM.

У широкомовних мережах множинного доступу, таких як Ethernet, може з'явитися велике число сусідів, тому потрібно призначати маршрутизатор DR.

У мережах типу точка-точка встановлення відносин повної суміжності не є складним, оскільки, за визначенням, у цих мережах на каналі знаходиться тільки два маршрутизатори. Призначення маршрутизатора DR не є обов'язковим і не виконується.

У мережах NBMA маршрутизатор OSPF може працювати в двох режимах:

– симульоване широкомовне середовище: адміністратор може призначити тип мережі як широкомовний, при цьому мережа імітує широкомовну модель шляхом призначення маршрутизатора DR чи BDR. У даному оточенні звичайно рекомендується, щоб адміністратор вибирав маршрутизатор DR чи BDR шляхом налаштування пріоритету маршрутизатора. Це забезпечує наявність у маршрутизатора DR чи BDR можливості повної передачі даних на всі сусідні маршрутизатори. Сусідні маршрутизатори також визначаються статично за допомогою команди neighbor у режимі налаштування OSPF;

– багатоточкове середовище: у даному середовищі кожна неширокомовна мережа розглядається як набір двоточкових каналів, при цьому маршрутизатор DR не призначається. Для цього середовища також потрібно, щоб сусідні маршрутизатори призначалися статично.

Області OSPF

Усі мережі OSPF починаються з області 0, яку називають областю магістралі. В міру розширення мережі можуть створюватися інші області, суміжні з областю 0. Цим іншим областям може призначатися будь-який номер до 65535.

Мережа OSPF має двохшарову ієрархічну структуру. Область 0 знаходиться вгорі, а всі інші області розташовані на наступному рівні. Усі немагістральні області повинні прямо з'єднуватися з областю 0. Ця група областей створює автономну систему OSPF (AS).

Процес роботи протоколу OSPF у межах якої-небудь області відрізняється від процесів, що виконуються між цією областю й областю магістралі. Звичайно між областями відбувається підсумовування мережевої інформації. Це дозволяє зменшити розмір таблиць маршрутизації в опорній мережі. За допомогою підсумовування також ізолюються зміни і нестабільності, чи биття, каналів до окремої області в домені маршрутизації. Якщо використовується підсумовування, то при якій-небудь зміні в топології отримують оголошення LSA і виконують алгоритм SPF тільки маршрутизатори, що знаходяться в області, де відбулася зміна.

Маршрутизатор, через який будь-яка область з'єднується з областю магістралі, називається прикордонним маршрутизатором області (ABR). Маршрутизатор, через який яка-небудь область з'єднується з іншим протоколом маршрутизації, таким як EIGRP, статичними маршрутами, які перерозподіляються в області OSPF, називається прикордонним маршрутизатором автономної системи (ASBR).

Впровадження протоколу OSPF

Налаштування протоколу OSPF в одній області [12]

Налаштування базового протоколу OSPF не є складною задачею і складається тільки з двох кроків. Перший крок – включення процесу маршрутизації OSPF. Другий крок – визначення мереж, що повинні бути оголошені.

Крок 1. Включення OSPF

router(config)#router ospf <ідентифікатор процесу>

Ідентифікатор процесу вибирається адміністратором, він може являти собою будь-яке число в діапазоні від 1 до 65535. Ідентифікатор процесу має тільки локальне значення і необов'язково повинен збігатися з ідентифікатором інших маршрутизаторів OSPF.

Крок 2. Оголошення мереж

Router(config-router)#network <адреса мережі> <шаблонна маска> area <ідентифікатор області>

Команда network має таку ж функцію, як в інших протоколах маршрутизації IGP. Цією командою визначаються інтерфейси, що можуть відправляти і приймати пакети OSPF. Дана інструкція визначає мережі, що включаються у оновлення маршрутизації OSPF.

У команді OSPF network використовується сполучення мережевої адреси і шаблонної маски. Мережева адреса, поряд із шаблонною маскою, вказує адресу інтерфейсу, чи діапазон адрес, що буде включений для OSPF.

Ідентифікатор області визначає область OSPF, що належить мережі. Навіть, якщо ніякі області не зазначені, повинна бути присутня яка-небудь область 0. В середовищі OSPF з однією областю, область завжди має ідентифікатор 0.

Команда OSPF network повинна використовувати шаблонні маски. При використанні для підсумовування мереж, чи організації супермереж, шаблонна маска є зворотньою мережевою маскою.

Щоб визначити шаблонну маску для підмережі, потрібно відняти десяткову мережеву маску для інтерфейсу від маски з усіма цифрами 255 (255.255.255.255).

Візьмемо наступний приклад: адміністратор хоче оголосити мережу 10.10.10.0/24 у протоколі OSPF. Маскою мережі для даного інтерфейсу Ethernet буде /24 чи 255.255.255.0. Потрібно відмінусувати маску від маски з усіма цифрами 255 для отримання шаблонної маски.

Отримане вираження OSPF network має вигляд:

Router(config-router)#network 10.10.10.0 0.0.0.255 area 0

Налаштування аутентифікації OSPF

Як і в інших протоколах маршрутизації, при налаштуванні за замовчуванням у протоколі OSPF обмін інформацією між сусідніми маршрутизаторами виконується простим текстом. У зв'язку з цим з'являється можлива загроза безпеці мережі.

Щоб усунути цю можливу проблему безпеки, потрібно налаштувати аутентифікацію OSPF між маршрутизаторами. Коли в якій-небудь області включена аутентифікація, маршрутизатори обмінюються інформацією тільки при співпадінні параметрів аутентифікації.

Використовуючи простий протокол аутентифікації по паролю, потрібно встановити на кожному маршрутизаторі ключ. Даний метод забезпечує тільки безпеку основного рівня, оскільки ключ передається між маршрутизаторами у вигляді простого тексту. Побачити ключ так просто, як і сам текст.

Більш безпечним методом аутентифікації є Message Digest 5 (MD5). При цьому методі на кожному маршрутизаторі необхідні ключ та ідентифікатор ключа. Маршрутизатор використовує алгоритм, за допомогою якого виконується обробка ключа, пакету OSPF та ідентифікатору ключа для створення зашифрованого числа. Зашифроване число міститься в кожному пакеті OSPF. Для отримання даного ключа неможливо використовувати програму перехоплення пакетів, оскільки ключ ніколи не передається.

Налаштування параметрів OSPF

Крім виконання базового налаштування протоколу OSPF потрібно налаштувати деякі його параметри.

Прикладом може служити ситуація, коли потрібно вказати, які маршрутизатори отримають функції маршрутизатора DR і BDR. Ця задача виконується шляхом встановлення пріоритету або інтерфейсу ідентифікатора маршрутизатора.

Маршрутизатор вибирає призначений маршрутизатор на підставі вищого значення кожного з наступних параметрів, у такій послідовності:

1. Пріоритет інтерфейсу. Встановлюється командою priority.

2. Ідентифікатор маршрутизатора. Задається командою OSPF router-id.

3. Вища адреса loopback інтерфейсу. За замовчуванням петлевий інтерфейс з вищою IP-адресою використовується як ідентифікатор маршрутизатора. Протокол OSPF підтримує петлеві інтерфейси, оскільки вони є не фізичними, а логічними. Логічні інтерфейси завжди мають пріоритет.

4. Вища адреса фізичного інтерфейсу. Як ідентифікатор маршрутизатора, маршрутизатор використовує вищу активну IP-адресу одного зі своїх інтерфейсів. Ця можливість викликає проблему, якщо інтерфейси припиняють роботу чи переналаштовуються.

Після зміни ідентифікатора маршрутизатора або пріоритету інтерфейсу, необхідно скинути значення відносин суміжності сусідніх маршрутизаторів. Для цього використовується команда clear ip ospf process. Цією командою вводяться в дію нові значення.

Ще одним параметром, що вимагає зміни, є пропускна здатність. На маршрутизаторах Cisco пропускна здатність більшості послідовних інтерфейсів за замовчуванням установлюється на швидкість 1,544 Мбіт/с, що відповідає швидкості стандарту T1. Значення пропускної здатності визначає вартість каналу, але фактично не впливає на швидкість каналу.

У деяких випадках провайдер послуг надає організації частину каналу T1, наприклад 384 Кбіт/с. Програмне забезпечення IOS робить припущення, що послідовні канали мають пропускну здатність T1, незважаючи на те, що інтерфейс фактично передає та отримує дані тільки зі швидкістю 384 Кбіт/с. Це припущення приводить до неправильного вибору шляху, оскільки протокол маршрутизації вирішує, що канал має більшу швидкість, ніж насправді.

Коли який-небудь послідовний інтерфейс фактично не працює зі швидкістю за замовчуванням T1, його потрібно налаштувати вручну (з обох сторін каналу на однакове значення).

У протоколі OSPF при налаштуванні інтерфейсу з використанням команди bandwidth чи ip ospf cost досягається однаковий результат. За допомогою обох команд вказується точне значення, яке OSPF використовує для визначення оптимального шляху.

Командою bandwidth змінюється значення пропускної здатності, що використовується для розрахунку метрики вартості протоколу OSPF. Щоб змінити вартість інтерфейсу, використовується команда ip ospf cost.

Ще одним параметром, зв'язаним з метрикою вартості OSPF, є еталонна пропускна здатність, що використовується для розрахунку вартості інтерфейсу, яку також називають вартістю каналу.

При розрахунку значення пропускної здатності кожного інтерфейсу використовується рівняння 100 000 000/пропускна здатність. Значення 100 000 000, чи 10^8, називається еталонною пропускною здатністю.

Існує певна складність з більш швидкісними каналами, такими як канали Gigabit Ethernet і 10Gbit Ethernet. Використання еталонної пропускної здатності за замовчуванням 100 000 000 дозволяє отримати інтерфейси зі значеннями пропускної здатності 100 Мбіт/с і вище, а також з однаковою вартістю OSPF зі значенням 1.

Для отримання більш точних розрахунків вартості може виникнути потреба підкоректувати значення еталонної пропускної здатності. Еталонна пропускна здатність змінюється використанням команди OSPF auto-cost reference-bandwidth.

При необхідності, ця команда, використовується на всіх маршрутизаторах, щоб метрика маршрутизації OSPF залишалася єдиною. Нова еталонна пропускна здатність вказується в мегабітах за секунду (Мбіт/с). Щоб встановити еталонну пропускну здатність на швидкість 10 Гбіт/с, використовується значення 10000.

Перевірка роботи протоколу OSPF

Після налаштування протоколу OSPF для перевірки правильності його роботи можна скористатися декількома командами.

Щоб у процесі пошуку та усунення несправностей у мережах OSPF перевірити, чи маршрутизатор створив відношення суміжності із сусідніми маршрутизаторами, використовується команда show ip ospf neighbor.

Якщо ідентифікатор сусіднього маршрутизатора не відображається або, якщо він не показує стан FULL, то обидва маршрутизатори не створили відносини суміжності OSPF. У випадку з маршрутизатором DROther відношення суміжності створене, якщо показується стан FULL чи 2WAY.

У мережі Ethernet із множинним доступом після стану FULL/ у колонку State (стан) відображаються значки DR і BDR.

Використання декількох протоколів маршрутизації

Налаштування і поширення маршруту за замовчуванням

Більшість мереж з'єднуються одна з одною по мережі Інтернет. Маршрутизатор OSPF надає інформацію маршрутизації про мережі в межах однієї автономної системи. Маршрутизатор OSPF також повинен надавати інформацію про доступність мереж за межами автономної системи.

Іноді адміністратори налаштовують статичні маршрути на маршрутизаторах таким чином, щоб вони надавали інформацію, що не виходить через протокол маршрутизації. Налаштування статичних маршрутів на всіх маршрутизаторах у великій мережі – процес трудомісткий. Більш легкий спосіб – налаштування маршруту за замовчуванням, що вказує на підключення до якої-небудь мережі через Інтернет.

За допомогою маршрутизатора OSPF адміністратор налаштовує даний маршрут на прикордонному маршрутизаторі автономної системи (ASBR). Маршрутизатор ASBR також часто називають маршрутизатором границі автономної системи. Маршрутизатор ASBR з'єднує мережа OSPF з якою-небудь зовнішньою мережею. Як тільки маршрут за замовчуванням буде внесений у таблицю маршрутизатора ASBR, його можна налаштувати таким чином, щоб він повідомляв цей маршрут для іншої частини мережі OSPF (default-information originate). Цей процес повідомляє маршрут за замовчуванням кожному маршрутизатору в межах автономної системи, що рятує адміністратора від роботи з налаштування статичних маршрутів на кожному маршрутизаторі в мережі.

Для налаштування маршрутизатора на розсилання маршруту за замовчуванням по мережі OSPF потрібно виконати наступні два кроки [12].

Крок 1

Налаштувати маршрут за замовчуванням на маршрутизаторі ASBR.

R1(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0

У команді статичного маршруту за замовчуванням можна вказати який-небудь інтерфейс чи IP-адресу наступного переходу.

Крок 2

Налаштувати на маршрутизаторі ASBR розсилання маршруту за замовчуванням на інші маршрутизатори. За замовчуванням маршрутизатор OSPF не включає маршрут за замовчуванням у свої оголошення навіть тоді, коли цей маршрут присутній у його таблиці маршрутизації.

R1(config)#router ospf 1

R1(config-router)#default-information originate

Тепер у таблицях маршрутизації інших маршрутизаторів в області OSPF має бути записаний шлюз останньої надії і вхід у мережу 0.0.0.0/0. Маршрут за замовчуванням включається в область OSPF і виглядає в таблицях маршрутизації інших маршрутизаторів просто як E2.

Налаштування підсумовування OSPF

Одним з методів, що забезпечують скорочення кількості оновлень маршрутів і розміру таблиці маршрутів OSPF, є підсумовування маршрутів. Маршрути можна підсумовувати в OSPF або між областями, що входять у ту ж мережу OSPF.

Коли маршрутизатор використовує сумарний маршрут, він використовує одну супермережеву адресу для представлення декількох маршрутів. Для оголошення сервером сумарного маршруту фактично активним має бути хоча б один з маршрутів, включених у сумарний маршрут.

Якщо нестабільні один чи декілька маршрутів, маршрутизатор продовжить повідомляти тільки стабільний сумарний маршрут. Він не відправляє оновлених даних про окремі маршрутизатори. Усі пакети, спрямовані на нестабільний маршрут під час його неактивного стану, будуть просто скидатися на підсумовуючий маршрутизатор.

Щоб налаштувати маршрутизатор OSPF ASBR на підсумовування цих сегментів мережі в іншу область OSPF, у режимі налаштування маршрутизатора використовується команда, яка вказує область, де сумуються мережі, а також початковий номер мережі та маску:

area ідентифікатор_області range ip-адреса маска_ip-адреси

Обмеження протоколу OSPF

Протокол OSPF є масштабованим протоколом маршрутизації. Він має можливість швидкої конвергенції та роботи з дуже великими мережами. Однак при його використанні необхідно звертати увагу на деякі обмеження.

Протокол OSPF підтримує кілька баз даних, тому вимагає більше пам’яті та обчислювальних потужностей, ніж протоколи маршрутизації на базі векторів відстані.

Алгоритм Дейкстри вимагає великої обчислювальної потужності для розрахунку найкращого маршруту. Якщо мережа OSPF є складною та нестабільною, даний алгоритм споживає значні об’єми ресурсів при частих перерахунках. Маршрутизатори, на яких виконується протокол OSPF, звичайно є більш потужними та дорогими.

Щоб уникнути надмірного використання ресурсів маршрутизатора, потрібно використовувати строгу ієрархічну структуру для поділу мережі на області меншого розміру. Всі області повинні підтримувати можливість зв’язку з областю 0. У протилежному випадку вони можуть втратити зв’язок з іншими областями.

Якщо мережа має великий розмір і складну структуру, процес налаштування протоколу OSPF може виявитися складним. Крім того, для інтерпретації інформації, що знаходиться в базах даних OSPF і таблицях маршрутизації, потрібно добре розуміння процесу.

Під час процесу початкового виявлення мережі протокол OSPF може виконувати лавинне розсилання по мережі повідомлень про стан каналу, істотно обмежуючи об’єм даних, що можуть бути передані по мережі. Лавинне розсилання у великих мережах з великою кількістю маршрутизаторів і маленькою смугою пропускання приводять до істотного зниження пропускної здатності мережі.

Незважаючи на проблеми і обмеження OSPF, він як і раніше є найбільш вживаним протоколом маршрутизації на базі стану каналу в середніх та великих корпоративних мережах.

Використання декількох протоколів

З багатьох причин організації можуть вибирати кілька протоколів маршрутизації.

Для різних розділів мережі мережевий адміністратор може вибрати різні протоколи маршрутизації в залежності від наявного застарілого обладнання та доступних ресурсів.

Можливі випадки злиття двох компаній, мережі яких були налаштовані з використанням різних протоколів маршрутизації, при цьому їм потрібно зв’язуватися один з одним по мережі.

Якщо на одному маршрутизаторі є кілька протоколів маршрутизації, то, можливо, що цей маршрутизатор буде дізнаватися про якого-небудь адресата з декількох джерел. Для маршрутизатора необхідний передбачуваний спосіб вибору кращого маршруту і запису його в таблицю маршрутизації.

Коли маршрутизатор довідається інформацію про одну мережу з декількох джерел, для визначення маршруту він використовує адміністративну відстань (Administrative Distance, AD). Всім методам отримання інформації про маршрутизацію програмне забезпечення Cisco IOS призначає визначену адміністративну відстань.

Якщо маршрутизатор дізнається про конкретну мережу за допомогою протоколів RIP та OSPF, для таблиці маршрутизації він вибере маршрут, повідомлений по протоколу OSPF. Його адміністративна відстань менша і тому більш бажана. Код на початку таблиці маршрутизації вказує джерело маршруту, або яким чином він був повідомлений. Кожен код асоціюється з конкретною адміністративною відстанню.

Якщо дві мережі мають однакову базову адресу і мережеву маску, маршрутизатор розглядає їх як ідентичні. Він розглядає підсумовану мережу та окрему мережу, що входить у цю сумарну мережу, як різні мережі.

Підсумована мережа 192.168.0.0/22 і окрема мережа 192.168.1.0/24 є окремими записами, незважаючи на те, що підсумована мережа містить цю окрему мережу. При виникненні цієї ситуації в таблицю маршрутизації заносяться обидві мережі. Вибір маршруту випадає на запис з найдовшим збігом префіксу.

Наприклад, маршрутизатор отримує пакет з IP-адресою призначення 172.16.0.10. Цьому пакету підходить три можливих маршрути: 172.16.0.0/12, 172.16.0.0/18 і 172.16.0.0/26. З усіх трьох маршрутів маршрут з адресою 172.16.0.0/26 має найдовший збіг. Щоб ці маршрути розглядалися як придатні, в мережевій масці маршруту повинна співпадати деяка кількість бітів.

Тема 5. Протоколи канального, мережевого та транспортного рівнів

Значення протоколів у моделі OSI та TCP/IP. Взаємодія рівнів і роль інкапсуляції даних. Протоколи канального рівня: основні функції канального рівня (адресація, контроль доступу до середовища, виявлення помилок). Ethernet як домінуючий стандарт локальних мереж. Технології бездротового доступу (Wi-Fi – IEEE 802.11). PPP та HDLC у каналах точка-точка. VLAN (IEEE 802.1Q) як розширення можливостей канального рівня. Протоколи мережевого рівня: призначення мережевого рівня (маршрутизація, логічна адресація). IPv4 та IPv6: структура адрес, основні відмінності. Протокол ARP і його роль у відображенні адрес. ICMP як інструмент діагностики та повідомлення про помилки.

Протоколи транспортного рівня: функції транспортного рівня (управління з'єднанням, контроль потоку, надійність). Порівняння TCP та UDP у прикладних сценаріях. Порти та сокети як механізм взаємодії з прикладними сервісами. Взаємодія протоколів різних рівнів. Приклади проходження даних крізь стек протоколів (інкапсуляція/декапсуляція). Приклади формування кадру, пакета та сегмента. Типові проблеми (затримки, втрата пакетів, дублювання) та методи їх усунення. Значення узгодженої роботи протоколів для функціонування глобальних мереж.

Значення протоколів у моделях OSI та TCP/IP

Протоколи є формалізованими правилами взаємодії між мережевими пристроями, які визначають формат даних, порядок обміну, способи виявлення та обробки помилок, а також механізми керування передаванням. Саме протоколи забезпечують сумісність обладнання та програмного забезпечення різних виробників і роблять можливим функціонування глобальних комп'ютерних мереж.

Еталонна модель OSI (Open Systems Interconnection) описує процес передавання даних у вигляді семирівневої ієрархії, де кожен рівень виконує чітко визначені функції та надає сервіси вищому рівню. Модель TCP/IP, що лежить в основі Інтернету, є більш практично орієнтованою і складається з чотирьох рівнів, які агрегують функції моделі OSI.

Попри різну кількість рівнів, обидві моделі ґрунтуються на однакових принципах: розподілі функцій, модульності та ієрархічній взаємодії протоколів.

Взаємодія рівнів і роль інкапсуляції даних

Передавання даних у мережі здійснюється завдяки процесу інкапсуляції. Дані, сформовані на прикладному рівні, послідовно передаються нижчим рівням, де кожен рівень додає власну службову інформацію у вигляді заголовка (а в деяких випадках і трейлера). У результаті на фізичному рівні передається бітовий потік.

На приймальному боці виконується зворотний процес − деінкапсуляція, під час якого кожен рівень аналізує та вилучає власний заголовок, передаючи корисні дані вищому рівню.

Такий підхід забезпечує:

– незалежність реалізації протоколів на різних рівнях;

– можливість модернізації окремих рівнів без зміни всієї системи;

– масштабованість і гнучкість мереж.

Канальний рівень: функції та протоколи

Канальний рівень відповідає за надійне передавання кадрів між безпосередньо з'єднаними вузлами мережі. Основні його функції полягають у фізичній адресації, контролі доступу до середовища передавання та виявленні помилок. На цьому рівні використовується MAC-адресація, яка ідентифікує мережеві інтерфейси. Контроль доступу до середовища (MAC − Media Access Control) визначає, який пристрій і в який момент часу має право передавати дані, що особливо важливо в мережах зі спільним середовищем. Для виявлення спотворень даних застосовуються контрольні суми та циклічні надлишкові коди (CRC), які дозволяють приймальній стороні визначити факт помилки та ініціювати повторне передавання кадру.

Ethernet (IEEE 802.3) є найпоширенішою технологією локальних мереж. Його домінування зумовлене простотою реалізації, високою масштабованістю та постійним зростанням пропускної здатності − від 10 Мбіт/с до сотень гігабіт за секунду.

Кадр Ethernet містить MAC-адреси відправника й отримувача, поле типу протоколу вищого рівня та контрольну суму. Сучасні Ethernet-мережі використовують комутацію замість спільного середовища, що практично усуває колізії.

Бездротові технології доступу: Wi-Fi (IEEE 802.11)

Стандарт IEEE 802.11 (Wi-Fi) реалізує канальний і фізичний рівні для бездротових локальних мереж. На відміну від Ethernet, Wi-Fi використовує механізми уникнення колізій (CSMA/CA), оскільки пристрої не завжди можуть «чути» один одного. Wi-Fi забезпечує мобільність користувачів, але є більш чутливим до завад, затримок і проблем безпеки, що зумовлює використання механізмів шифрування, автентифікації та керування доступом.

PPP та HDLC у каналах «точка-точка»

Протоколи PPP (Point-to-Point Protocol) та HDLC (High-Level Data Link Control) застосовуються в каналах з'єднання «точка-точка». Вони забезпечують інкапсуляцію мережевих протоколів, перевірку цілісності даних та, у випадку PPP, механізми автентифікації (PAP, CHAP). PPP широко використовується у з'єднаннях через WAN та VPN, тоді як HDLC часто застосовується у провайдерських мережах.

VLAN (IEEE 802.1Q) як розширення канального рівня

VLAN (Virtual Local Area Network) дозволяє логічно сегментувати фізичну мережу на окремі віртуальні домени широкомовлення. Стандарт IEEE 802.1Q реалізує тегування кадрів Ethernet, що дає змогу одному фізичному каналу обслуговувати декілька VLAN. Використання VLAN підвищує безпеку, керованість і ефективність використання мережевих ресурсів, особливо в корпоративних мережах.

Протоколи мережевого рівня

Мережевий рівень відповідає за логічну адресацію та маршрутизацію пакетів між різними мережами. IPv4 використовує 32-бітну адресу, що обмежує адресний простір і потребує застосування NAT. IPv6 використовує 128-бітну адресу, що практично усуває проблему дефіциту адрес і спрощує маршрутизацію, автоконфігурацію та безпеку. Основні відмінності між IPv4 та IPv6 полягають у структурі заголовка, механізмах адресації та підтримці сучасних мережевих сервісів.

ARP і ICMP

Протокол ARP (Address Resolution Protocol) забезпечує відображення логічних IP-адрес у фізичні MAC-адреси в локальній мережі.

ICMP (Internet Control Message Protocol) використовується для діагностики та повідомлення про помилки, зокрема у таких утилітах, як ping і traceroute.

Транспортний рівень: TCP та UDP

Транспортний рівень забезпечує логічний зв'язок між прикладними процесами.

TCP (Transmission Control Protocol) гарантує надійне передавання даних, контроль потоку та впорядкування сегментів, що робить його придатним для веб-сервісів, електронної пошти та передавання файлів.

UDP (User Datagram Protocol) є ненадійним, але швидким, що робить його ефективним для потокового відео, VoIP і онлайн-ігор.

Порти та сокети дозволяють ідентифікувати конкретні прикладні сервіси на вузлі та забезпечують мультиплексування з'єднань.

Проходження даних крізь стек протоколів

У процесі передавання інформації прикладні дані інкапсулюються у сегмент транспортного рівня, який далі поміщається в пакет мережевого рівня, а потім − у кадр канального рівня. На фізичному рівні кадр перетворюється на послідовність бітів.

Наприклад, HTTP-повідомлення інкапсулюється у TCP-сегмент, який передається у складі IP-пакета, що, своєю чергою, поміщається в Ethernet-кадр.

Типові проблеми передавання даних і методи їх усунення

До основних проблем належать затримки, втрата пакетів, дублювання та зміна порядку доставки. Вони усуваються завдяки механізмам повторного передавання, керування потоком, буферизації та маршрутизації.

Функціонування глобальних мереж можливе лише за умови узгодженої роботи протоколів усіх рівнів. Порушення або некоректна реалізація хоча б одного рівня призводить до деградації продуктивності або повної недоступності сервісів

Тема 6. Віртуальні локальні мережі та віртуальні приватні мережі

Актуальність технологій віртуалізації мережевих ресурсів. Віртуальні локальні мережі (VLAN): основні поняття та принципи роботи. Переваги використання VLAN: ізоляція трафіку, підвищення продуктивності та безпеки. Типи VLAN (на основі портів, MAC-адрес, протоколів). Механізм тегування кадрів (IEEE 802.1Q). Приклади конфігурації VLAN у мережевому обладнанні (комутатори Cisco, Mikrotik).

Віртуальні приватні мережі (VPN): основні поняття та призначення VPN. Основні архітектури VPN: site-to-site та remote access. Протоколи VPN (PPTP, L2TP, IPSec, SSL/TLS, OpenVPN, WireGuard). Використання VPN у корпоративних та хмарних середовищах. Використання VLAN і VPN для забезпечення сегментації та захисту корпоративних мереж. Практичні аспекти реалізації: налаштування VLAN у Cisco Packet Tracer. Створення простого VPN-з'єднання між віддаленими офісами. Типові проблеми та шляхи їх вирішення. Значення VLAN і VPN у сучасних інформаційно-комунікаційних системах. Перспективи розвитку технологій віртуалізації та безпечного віддаленого доступу.

Вузли і сервери, підключені до комутаторів 2-го рівня, вважаються частиною мережевого сегмента. Така організація характеризується двома серйозними проблемами:

– комутатори виконують лавинне розсилання широкомовних кадрів з усіх портів, що приводить до невиправданого використання смуги пропускання. Зі збільшенням числа пристроїв, підключених до комутатора, генерується більше широкомовного трафіку;

– всі пристрої, підключені до комутатора, можуть пересилати та отимувати кадри від всіх інших пристроїв на цьому комутаторі.

При проектуванні мережі рекомендується обмежувати широкомовний трафік областю мережі, у якій він необхідний. Існують причини організаційного характеру, коли одні вузли можуть отримувати доступ один до одного, а інші ні. Для обмеження широкомовних розсилань і об'єднання вузлів у групи створюються віртуальні локальні мережі (Virtual Local Area Networks, VLAN) [11].

VLAN – це логічний домен широкомовного розсилання, що може охоплювати кілька фізичних сегментів LAN. VLAN дозволяє адміністратору поєднувати станції по логічній функції незалежно від фізичного положення користувачів.

Кожна VLAN функціонує як окрема локальна мережа. VLAN може охоплювати один чи кілька комутаторів, що дозволяє вузлам працювати так, ніби вони знаходилися в одному сегменті.

VLAN виконують такі функції:

– обмеження широкомовних розсилань;

– об'єднання пристроїв у групи;

– пристрої, розташовані в одній VLAN, невидимі для пристроїв, розташованих в інший VLAN.

Для передачі трафіку між VLAN необхідний пристрій 3-го рівня.

Основними перевагами використання віртуальних локальних мереж є:

– безпека – віртуальні мережі дозволяють відокремити групи, які мають важливі дані від загальної частини мережі, зменшуючи ймовірність втрат конфіденційної інформації;

– зниження витрат – економія за рахунок ефективнішого використання пропускної здатності, up-лінків, зменшення потреб модернізації мережі;

– збільшення продуктивності – поділ мереж 2-го рівня на кілька логічних робочих груп (широкомовних доменів) зменшує зайвий трафік у мережі та підвищує продуктивність роботи;

– зменшення широкомовного шторму – поділ мережі на VLAN, знижує кількість пристроїв, які можуть брати участь в поширенні широкомовного шторму. Сегментація мережі запобігає поширенню широкомовного шторму на всю мережу;

– покращення ефективності роботи IT-персоналу – використання VLAN спрощує управління мережею, оскільки користувачі зі схожими вимогами до мережевих ресурсів знаходяться в одній VLAN. При додаванні нового комутатора всі політики і процедури, вже

налаштовані для конкретної VLAN, застосовуються при активації порта. Для певної VLAN можна задати відповідне ім’я з врахуванням їх призначення.

Пристрій можна призначити у VLAN відповідно до його розташування, MAC-адреси, IP-адреси або прикладних програм, які він використовує найчастіше. Адміністратори задають приналежність пристрою до VLAN статично або динамічно.

У випадку статичного призначення адміністратор повинен вручну призначити кожен порт комутатора у визначену VLAN. Наприклад, порт fa0/3 можна призначити в VLAN 20. Будь-який пристрій, що підключається до порту fa0/3, автоматично стає членом VLAN 20. Цей тип приналежності до VLAN найлегше налаштувати і він найпопулярніший, але додавання, переміщення і зміна пристроїв потребує постійного втручання адміністратора. Наприклад, переміщення вузла з однієї VLAN у другу потребує або ручного перепризначення порту комутатора в нову VLAN, або переключення кабелю робочої станції в інший порт комутатора, що відноситься до нової VLAN. Приналежність пристрою до VLAN цілком прозора для користувачів. Користувачі, які працюють із пристроєм, підключеним до порту комутатора, не знають, до якої VLAN вони належать.

Динамічна приналежність VLAN вимагає наявності сервера політик VLAN (VMPS − VLAN Management Policy Server). VMPS містить базу даних, що співставляє MAC-адреси з VLAN мережами. Коли пристрій підключається до порту, VMPS шукає його MAC-адресу у своїй базі даних і тимчасово призначає порт у відповідну VLAN. Динамічна приналежність VLAN вимагає більш складного налаштування й організації, але формує більш гнучку структуру, ніж статична приналежність VLAN. Переміщення, додавання і зміна компонентів виконується автоматично і не вимагає втручання адміністратора. Не всі комутатори Catalyst підтримують VMPS.

Типи VLAN

В мережі може існувати декілька типів VLAN. Деякі з них визначаються типом трафіку, що передається, інші визначаються функціями, що виконуються. Наприклад, у випадку статичного призначення порту у визначену VLAN, така VLAN називатиметься VLAN доступу(access VLAN).

Загалом розрізняють такі типии VLAN:

– Data VLAN;
– Default VLAN;
– Native VLAN;
– Management VLAN;
– Voice VLANs.

Data VLAN

VLAN даних – це VLAN налаштована для передавання лише користувацького трафіку. VLAN може передавати голосовий трафік, трафік для управління комутатором, але цей трафік не буде частиною Data VLAN. Це звичайна практика, щоб відокремити голосовий трафік та трафік управління від трафіку даних. VLAN даних іноді називають користувацькою VLAN.

Default VLAN

Всі порти комутатора стають членом VLAN за замовчуванням після початкового завантаження комутатора. Усі порти комутатора будучи членами Default VLAN знаходяться в одному домені широкомовної передачі. Це дозволяє будь-якому пристрою, підключеному до будь-якого порту комутатора, зв’язатись з іншими пристроями на інших портах комутатора. Для комутаторів Cisco Default VLAN є VLAN 1. VLAN 1 має всі функції VLAN, за винятком того, що її не можна перейменувати чи видалити. За замовчуванням, трафік 2-го рівня, такий як CDP чи STP пов’язаний з VLAN 1. З метою підвищення безпеки рекомендується змінити Default VLAN з VLAN 1 на будь-яку іншу VLAN, що в свою чергу вимагатиме налаштування всіх портів комутатора

Native VLAN

Native VLAN – це VLAN, який отримує всі кадри, які надсилаються без тега, або кадри з нерозподілених портів (не включених в жоден VLAN). 802.1q транковий порт підтримує трафік з багатьох VLAN (тегований трафік), а також трафік, який надходить не з

VLAN (нетегований трафік). Якщо комутатор отримує нетеговані кадри на транковому порті він автоматично передає їх в Native VLAN. В ролі Native VLAN рекомендується використовувати VLAN 1.

В термінології інших виробників можуть вживатися інші назви для такої VLAN. Наприклад untagged VLAN (3Com, Planet, D-link, Zyxel, HP).

Management VLAN

VLAN керування може бути будь-яка VLAN налаштована для доступу до функцій управління комутатором. VLAN 1 буде функціонувати як VLAN управління, якщо не призначити іншу VLAN в якості Management VLAN. Для VLAN керування призначається IP-адреса та маска підмережі. Керувати комутатором можна через HTTP, Telnet, SSH або SNMP.

Voice VLANs

Для передачі голосу по IP (VoIP) рекомендовано використовувати окремий VLAN, оскільки VoIP-трафік вимагає:

– гарантовану смуга пропускання для забезпечення якості передачі голосу;

– пріоритетності обслуговування в порівнянні з іншими типами трафіку;

– затримки менше 150 мілісекунд (мс) в мережі.

Для задоволення цих вимог вся мережа повинна бути призначена для підтримкиVoIP.

Ідентифікація та налаштування VLAN

Максимальне загальне число статичних і динамічних VLAN залежить від типу комутатора і версії IOS. За замовчуванням у якості VLAN керування застосовується VLAN1. Адміністратори використовують IP-адресу VLAN керування для віддаленого налаштування комутатора. Віддалений доступ до комутатора дозволяє адміністратору мережі налаштовувати та обслуговувати всі конфігурації VLAN. Крім того, VLAN керування використовується для обміну даними, наприклад трафіком протоколів CDP (Cisco Discovery Protocol) і VTP (VLAN Trunking Protocol), з іншими мережевими пристроями. При створенні віртуальної мережі їй призначається номер та ім’я. Номер VLAN – це будь-яке число з діапазону, доступного комутатору, крім VLAN1. Різні комутатори підтримують різну кількість VLAN. Найменування VLAN вважається рекомендованим методом керування.

Для налаштування VLAN використовуються наступні команди режиму глобальної конфігурації:

Switch(config)#vlan номер_vlan

Switch(config-vlan)#name ім’я_vlan

Switch(config-vlan)#exit

За замовчуванням всі порти належать до VLAN1. Порти можна призначати по одному чи діапазонами.

Для призначення окремих портів використовуються наступні команди:

Switch(config)#interface fa0/номер_порту

Switch(config-if)#switchport access vlan номер_vlan

Switch(config-if)# exit

Для призначення діапазонів портів використовуються наступні команди:

Switch(config)#interface range fa0/початок_діапазону - кінець_діапазону

Switch(config-if)#switchport access vlan номер_vlan

Switch(config-if)#exit

Для перевірки, обслуговування та усунення несправностей VLAN важливо знати основні команди show, доступні в Cisco IOS.

Для перевірки й обслуговування VLAN використовуються наступні команди:

– show vlan − виводить докладний список номерів та імен VLAN, активних на комутаторі, а також портів, призначених у кожну з них;

– show vlan brief − виводить зведений список, у якому відображаються тільки активні VLAN і їхні порти.

– show vlan id номер_ідентифікатора − виводить звіт про визначену VLAN за її ідентифікатором;

– show vlan name ім’я_vlan − виводить звіт про визначену VLAN за її ім’ям.

В організації працівники часто приходять, звільняються або переміщуються між відділами і проектами. Цей постійний рух вимагає обслуговування VLAN, включаючи їхнє видалення і перепризначення портів. Видалення VLAN і перепризначення портів в інші VLAN – це дві різні функції. Коли порт видаляється з визначеної VLAN, він повертається у VLAN1. При видаленні VLAN усі пов’язані порти деактивуються, оскільки вони більш незв’язані з жодною VLAN.

Видалення VLAN:

Switch(config)#no vlan номер_vlan

Видалення порту з визначеної VLAN:

Switch(config)#interface fa0/номер_порту

Switch(config-if)#no switchport access vlan номер_vlan

Пристрої, підключені до VLAN, взаємодіють тільки з іншими пристроями в цій VLAN, при цьому пристрої можуть бути підключені як до одного, так і до різних комутаторів.

Комутатор зв’язує кожен порт із визначеним номером VLAN. При прийомі кадру на порт комутатор додає ідентифікатор VLAN (VI) у кадр Ethernet. Додавання ідентифікатора VLAN у кадр Ethernet називається маркуванням кадру. Найпоширеніший стандарт маркування кадру – IEEE 802.1q. Стандарт 802.1q, що іноді скорочується до dot1q, має на увазі вставку 4-байтного поля мітки в кадр Ethernet. Ця мітка знаходиться між адресою джерела і полем type/length (тип/довжина). Мінімальний розмір кадрів Ethernet складає 64 байти, максимальний – 1518 байтів, однак розмір тегованого кадру Ethernet може досягати 1522 байти.

Кадри включають такі поля:

– MAC-адреси джерела і призначення;

– довжина кадру;

– корисні дані;

– контрольна послідовність кадру (FCS).

Поле FCS забезпечує виявлення помилок і гарантує цілісність всіх бітів в кадрі.

Мітка збільшує мінімальний розмір кадру Ethernet з 64 до 68 байт. Максимальний розмір збільшується з 1518 до 1522 байт. Комутатор перерозраховує FCS, тому що кількість бітів в кадрі збільшується. Якщо порт, сумісний з 802.1q, підключений до іншого порту, також сумісного з 802.1q, дані маркування VLAN передаються між ними. Якщо підключений порт несумісний з 802.1q, мітка VLAN буде видалена перш, ніж кадр досягне середовища передачі. Якщо пристрій, або порт доступу без підтримки 802.1q отримує кадр 802.1q, то дані мітки ігноруються, а пакет комутується на рівні 2 як стандартний кадр Ethernet. Це дозволяє розміщувати на транковому маршруті 802.1q проміжні пристрої рівня 2, наприклад інші комутатори. Щоб обробити кадр із міткою 802.1q, пристрій повинен дозволяти MTU зі значенням 1522 або вище.

VLAN для IP-телефонії та безпровідного доступу

Головне призначення VLAN – поділ трафіку на логічні групи. Трафік з однієї VLAN не впливає на трафік в іншій VLAN. Середовище VLAN ідеальне для трафіку, чуттєвого до тимчасових затримок, наприклад голосового. Голосовий трафік повинен отримувати пріоритет над звичайним трафіком даних, щоб уникнути перерв і джитера при розмові. Надання виділеної VLAN для голосового трафіку дозволяє голосовому трафіку не конкурувати з даними за доступну смугу пропускання.

IP-телефон, як правило, включає два порти – один для голосу, інший для даних. Пакети, передані між IP-телефоном і комп’ютером, використовують той же фізичний канал і той же порт комутатора. Для сегментації голосового трафіку потрібно активувати голосовий VLAN на комутаторі.

Безпровідний доступ – інший тип трафіку, що використовує переваги VLAN. Безпровідний доступ по своїй природі небезпечний та вразливий до хакерських атак. VLAN, створені для безпровідного доступу, ізолюють деякі з потенційних проблем. Загроза цілісності безпровідній VLAN не вплине на інші VLAN в організації. Більшість безпровідних

середовищ поміщають користувачів у VLAN за межами брандмауера для підвищеної безпеки. Користувачі повинні пройти аутентифікацію, щоб отримати доступ до внутрішньої мережі з безпровідної мережі. Крім того, багато організацій надають гостьовий доступ у свою безпровідну мережу. Гостьові облікові записи надають усім тимчасові безпровідні послуги, такі як веб-доступ, електронна пошта, ftp і SSH. Кількість активних облікових записів обмежується. Гостьові облікові записи включаються в безпровідну VLAN або групуються у власні VLAN.

Методи роботи з VLAN

При якісному плануванні та проектуванні VLAN забезпечується безпека, заощаджується смуга пропускання та локалізується трафік у корпоративній мережі. Усі ці функції поєднуються для покращення продуктивності мережі.

Деякі з методів налаштування, рекомендовані для VLAN у корпоративній мережі наводяться нижче:

– організація розміщення серверів;
– відключення портів, що не використовуються;
– налаштування VLAN керування з номером, відмінним від 1;
– використання протоколу VTP;
– налаштування доменів VTP;
– перезавантаження нових комутаторів перед їхнім додаванням в існуючу мережу.

В той же час VLAN не є відповіддю на всі проблеми.

Неправильне впровадження VLAN може привести до зайвого ускладнення мережі, що стане причиною зниження продуктивності мережі.

Транкінг та маршрутизація між VLAN

Режими роботи портів у VLAN

VLAN виконують три основні функції:

– обмеження розміру доменів широкомовних розсилань;
– підвищення продуктивності мережі;
– підвищення безпеки.

Щоб повною мірою скористатися перевагами VLAN, необхідно поширити їх на кілька комутаторів. Для портів комутатора можна задати дві різні ролі. Порт може бути визначений як порт доступу, або як транковий порт.

Порт доступу належить лише до однієї VLAN. Як правило, окремі пристрої, такі як комп'ютери і сервери, підключаються до портів такого типу. Якщо кілька комп'ютерів підключаються до одного порту доступу через концентратор, то всі пристрої, підключені до концентратора, будуть належати до однієї VLAN.

Транкінговий порт – це канал типу «точка-точка» між комутатором та іншим мережевим пристроєм. Транкові підключення служать для передачі трафіку декількох VLAN через один канал і забезпечують їм доступ до всієї мережі. Транкові порти необхідні для передачі трафіку декількох VLAN між пристроями при з'єднанні двох комутаторів, комутатора і маршрутизатора, комутатора і мережевого адаптера вузла з підтримкою транкінгу 802.1q. Без транкових портів для кожної VLAN було б потрібне окреме з'єднання між комутаторами. Наприклад, корпорації з 10 VLAN буде потрібно 10 каналів зв'язку. При такій організації мережа не масштабується належним чином. Транкові канали дозволяють вирішити цю проблему за рахунок передачі трафіку декількох VLAN через один канал. Для передачі трафіку декількох VLAN через один канал необхідна їх ідентифікація. Транковий порт підтримує маркування кадрів, яке дозволяє додати до кадру дані VLAN.

IEEE 802.1q – стандартний і затверджений метод маркування кадрів. Корпорація Cisco розробила власний протокол маркування кадрів з назвою міжкомутаторний канал (ISL, Inter-Switch Link). Комутатори вищого класу, такі як Catalyst 6500, підтримують обидва протоколи маркування, однак більшість комутаторів LAN, таких як Catalyst 2960, підтримують тільки 802.1q.

Налаштування режимів роботи портів

За замовчуванням порти комутатора працюють у режимі доступу. Щоб налаштувати порт комутатора в якості транкового порту, використовуються такі команди:

Switch(config)#interface fa0/номер_порту

Switch(config-if)#switchport mode trunk

Switch(config-if)#switchport trunk encapsulation {dot1q | isl | negotiate}

Комутатори, що підтримують і 802.1q і ISL, вимагають останньої команди. Наприклад комутатор Catalyst 2960 не вимагає цієї команди, тому що підтримує тільки 802.1q.

Параметр узгодження використовується за замовчуванням на багатьох комутаторах Cisco. Він дозволяє пристрою автоматично виявляти тип інкапсуляції сусіднього комутатора.

Нові комутатори можуть виявляти тип каналу, заданий на протилежній стороні. У залежності від підключеного пристрою канал налаштовується в якості транкового порту або як порт доступу.

Switch(config-if)#switchport mode dynamic {desirable | auto}

У режимі desirable порт стає транковим, якщо порт на іншій стороні знаходиться в режимі trunk, desirable або auto.

У режимі auto порт стає транковим, якщо порт на іншій стороні знаходиться в режимі trunk чи desirable.

Щоб повернути транковий порт у режим доступу, використовуються такі команди:

Switch(config)#interface fa0/номер_порту

Switch(config-if)#no switchport mode trunk

або

Switch(config-if)#switchport mode access

Транкінг дозволяє декільком VLAN пересилати трафік між комутаторами, використовуючи один порт.

Транковий канал пропускає трафік з 4-байтним полем мітки в кадрі, якщо на обох сторонах налаштований протокол 802.1q. Мітка кадру містить ідентифікатор VLAN ID.

Коли комутатор отримує тегований кадр на транковому порту, він видаляє мітку перш, ніж переслати кадр із порту доступу. Комутатор пересилає кадр, тільки якщо порт доступу відноситься до тієї ж VLAN, що і тегований кадр. Однак деякі типи трафіку повинні проходити через канал 802.1q без ідентифікатора VLAN. Трафік без ідентифікатора VLAN називається нетегованим. Приклади нетегованого трафіку: CDP (Cisco Discovery Protocol), VTP і певні типи голосового трафіку. Нетегований трафік мінімізує затримку, пов’язану з перевіркою мітки ідентифікатора VLAN. Для підтримки нетегованого трафіку використовується спеціальна VLAN, що називається рідна (native VLAN).

Нетеговані кадри, прийняті на порту 802.1q, передаються у native VLAN. На комутаторах Cisco Catalyst в якості native VLAN за замовчуванням використовується VLAN 1. Будь-яку VLAN можна налаштувати в якості native VLAN. Native VLAN для транкового підключення 802.1q повинна бути однаковою на обох сторонах каналу. В протилежному випадку в топології STP можуть виникнути петлі. Для призначення ідентифікатора native VLAN фізичному інтерфейсу для транкового підключення 802.1q використовується команда:

Switch(config-if)#dot1q native vlan ідентифікатор_vlan

Хоча VLAN можуть охоплювати кілька комутаторів, тільки пристрої, що відносяться до однієї VLAN, можуть взаємодіяти один з одним. VLAN ізолюють визначені типи трафіку з міркувань безпеки. Для переміщення трафіку між VLAN необхідний пристрій мережевого рівня, що збільшує вартість впровадження і підвищує рівень затримок в мережі. Використання пристрою 3-го рівня для з’єднання між VLAN дозволяє адміністратору здійснювати контроль над трафіком, що передається з однієї VLAN в іншу.

Методи маршрутизації між VLAN

Один з методів маршрутизації між VLAN вимагає окремого підключення інтерфейсу до пристрою 3-го рівня для кожної VLAN. Інший метод з’єднання між VLAN вимагає функцій, що називаються субінтерфейсами. Субінтерфейси дозволяють логічно розділити один фізичний інтерфейс на кілька логічних шляхів. Для кожної VLAN налаштовується окремий шлях або субінтерфейс.

Взаємодія між VLAN з використанням субінтерфейсів вимагає налаштування як маршрутизатора, так і комутатора. Інтерфейс комутатора має бути налаштований в якості транкового каналу 802.1q.

Інтерфейс маршрутизатора (не нижче FastEthernet 100 Мбіт/с) має бути налаштований з підтримкою інкапсуляції 802.1q, крім того, для кожної VLAN налаштовується один субінтерфейс. Субінтерфейс дозволяє кожній VLAN мати власний логічний шлях і шлюз за замовчуванням до маршрутизатора. Вузол з передавальної VLAN пересилає трафік маршрутизатору, використовуючи шлюз за замовчуванням. Субінтерфейс VLAN визначає шлюз за замовчуванням для усіх вузлів цієї VLAN. Маршрутизатор визначає IP-адресу призначення і виконує пошук по таблиці маршрутизації. Якщо VLAN призначення відноситься до того ж комутатора, що і вихідна VLAN, маршрутизатор пересилає трафік назад до вихідного комутатора, використовуючи параметри субінтерфейсу та ідентифікатора VLAN призначення. Така конфігурація часто називається каскадом маршрутизаторів.

Якщо вихідний інтерфейс маршрутизатора сумісний з 802.1q, кадр зберігає 4-байтну мітку VLAN. Якщо вихідний інтерфейс несумісний з 802.1q, маршрутизатор відділяє мітку від кадру і повертає кадр в оригінальний формат Ethernet.

Налаштування маршрутизації між VLAN

Порядок налаштування маршрутизації між VLAN:

1. Налаштувати транковий порт на комутаторі.

Switch(config)#interface fa0/2

Switch(config-if)#switchport mode trunk

2. На маршрутизаторі налаштувати інтерфейс FastEthernet без IP-адреси та маски.

Router(config)#interface fa0/1

Router(config-if)#no ip address

Router(config-if)#no shutdown

3. На маршрутизаторі налаштувати один субінтерфейс з IP-адресою та маскою для кожної VLAN. Кожен субінтерфейс використовує інкапсуляцію 802.1q.

Router(config)#interface fa0/0.10

Router(config-subif)#encapsulation dot1q 10

Router(config-subif)#ip address 192.168.10.1 255.255.255.0

4. Перевірити конфігурацію і працездатність маршрутизації між VLAN за допомогою таких команд:

Switch#show trunk

Router#show ip interfaces

Router#show ip interfaces brief

Router#show ip route

Тема 7. Поняття мережної безпеки. Принципи роботи ACL

Роль ACL у забезпеченні контролю доступу та мережевої безпеки. Поняття та класифікація ACL. Визначення списків контролю доступу. Стандартні та розширені ACL. Нумеровані та іменовані ACL. Принципи функціонування ACL: логіка «permit» та «deny». Типові сценарії застосування ACL: фільтрація трафіку за IP-адресою джерела і призначення, контроль доступу за протоколами та портами, використання ACL для сегментації трафіку у VLAN, обмеження доступу до мережевих сервісів. Приклади налаштування ACL на мережевому обладнанні. ACL як базовий, але необхідний інструмент мережевої безпеки.

Фільтрація трафіку

Безпека в корпоративній мережі відіграє важливу роль. Важливо запобігти несанкціонованому доступу та захистити мережу від різного роду атак. У випадку несанкціонованого доступу зловмисники можуть змінити, знищити чи викрасти конфіденційні дані. DoS-атаки перешкоджають доступу легітимних користувачів до ресурсів.

Фільтрація трафіку дозволяє адміністратору контролювати трафік у різних сегментах мережі. Фільтрація являє собою процес аналізу вмісту пакету з метою дозволу або блокування його передачі. Фільтрація пакетів може бути простою чи складною і може забороняти або дозволяти трафік за такими критеріями, як: вихідна IP-адреса, IP-адреса призначення, MAC-адреса; протокол, тип додатку. Фільтрацію пакетів можна порівняти з фільтрацією небажаної електронної пошти. Багато поштових додатків дозволяють користувачу налаштовувати автоматичне видалення повідомлень, що приходять з визначеної вихідної адреси. Фільтрація пакетів може здійснюватися схожим чином шляхом налаштування маршрутизатора на визначення небажаного трафіку. Фільтрація пакетів дозволяє підвищити продуктивність мережі. Завдяки відхиленню небажаного чи забороненого трафіку близько до його джерела, трафік не передається по мережі і не споживає ресурси. Фільтрація пакетів, яку іноді називають статичною фільтрацією пакетів, контролює доступ до мережі шляхом аналізу вхідних і вихідних пакетів та їх передачі або блокування на основі встановлених критеріїв.

Маршрутизатор діє як фільтр пакетів, коли пересилає або забороняє передачу пакетів відповідно до правил фільтрації. Коли пакет прибуває на маршрутизатор, який здійснює фільтрацію пакетів, маршрутизатор аналізує певну інформацію із заголовку пакету та приймає рішення відповідно до правил фільтрації. Фільтрація пакетів здійснюється на мережевому рівні моделі OSI, або Інтернет-рівні моделі TCP/IP.

Маршрутизатор використовує правила, щоб визначити: передавати чи заборонити трафік залежно від IP-адрес джерела і призначення, порту джерела і порту призначення, протоколу пакету. Ці правила визначаються за допомогою списків контролю доступу, або ACL на основі:

– IP-адреси джерела;

– IP-адреси отримувача;

– типу ICMP повідомлення.

ACL може також проаналізувати інформацію вищого рівня та співставити зі своїми правилами. Інформація вищого рівня включає в себе:

– TCP / UDP порт джерела;

– TCP / UDP порт призначення.

До пристроїв, що найчастіше використовуються для фільтрації трафіку, входять:

– міжмережеві екрани, вбудовані в інтегровані маршрутизатори;

– виділені пристрої забезпечення безпеки;

– сервери.

Деякі з них фільтрують тільки трафік, що виникає у внутрішній мережі. Більш досконалі пристрої безпеки здатні розпізнавати і фільтрувати відомі типи атак із зовнішніх джерел.

Корпоративні маршрутизатори здатні розпізнавати шкідливий трафік і запобігати його проникненню в мережу та порушенню працездатності мережі. Практично всі маршрутизатори виконують фільтрацію трафіку по вихідних і кінцевих IP-адресах пакетів. Вони також фільтрують визначені додатки і протоколи, такі як IP, TCP, HTTP, FTP, Telnet та ін.

Списки контролю доступу

Одним з найбільш розповсюджених способів фільтрації трафіку є використання списків контролю доступу (ACL-списків). ACL-списки можна використовувати для керування вхідним і існуючим трафіком у мережі і його фільтрації.

ACL – це скрипт конфігурації маршрутизатора, який визначає, чи буде маршрутизатор дозволяти або забороняти передачу пакетів на основі критеріїв, які містяться в заголовку пакета. Списки ACL також використовуються для вибору типів трафіку, які будуть проаналізовані, передані чи оброблені іншими способами.

Кожен пакет проходить через інтерфейс з певними ACL. ACL перевіряються зверху вниз, рядок за рядком, перевіряючи співпадіння шаблону у пакетах. ACL забезпечує корпоративну політику безпеки, застосовуючи правила заборони/дозволу для визначення долі пакета. Розмір ACL-списку може змінюватись від однієї інструкції, по якій дозволяється або блокується трафік від одного джерела, до сотні інструкцій, що дозволяють або забороняють пакети з різних джерел. В основному, ACL-списки використовуються для визначення типів пакетів, які приймаються чи відхиляються.

ACL-списки визначають трафік для декількох цілей:

– вказання внутрішніх вузлів для NAT;

– виявлення і класифікації трафіку для забезпечення розширених можливостей, таких як QoS і організація черги;

– обмеження інформації про оновлення маршрутизації;

– контроль доступу віртуальних терміналів до маршрутизаторів.

За замовчуванням, маршрутизатор не має жодних списків ACL і тому не фільтрує трафік. Трафік, який надходить на маршрутизатор маршрутизується відповідно до таблиці маршрутизації. Якщо на маршрутизаторі не використовуються ACL списки, всі пакети проходять через маршрутизатор до наступного сегменту мережі.

Загалом на маршрутизаторі можуть застосовуватись три типи списків ACL, по одному відповідно до протоколу, до напрямку та до інтерфейсу:

– один ACL на протокол – для контролю потоків трафіку на інтерфейсі, тобто ACL повинні бути визначені для кожного протоколу на інтерфейсі;

– один ACL одному напрямку – ACL контролюють трафік в одному напрямку в один момент часу на інтерфейсі. Два роздільних списки контролю доступом мають бути створені для контролю вхідного та вихідного трафіку;

– один ACL на інтерфейс – ACL контролює трафіку для одного інтерфейсу.

Написання списків ACL може бути складним та комплексним завданням.

Використання ACL-списків може бути пов’язане з наступними потенційними проблемами:

– додаткове навантаження на маршрутизатор для перевірки всіх пакетів означає менший час на фактичне пересилання пакетів;

– погано організовані ACL-списки створюють ще більше навантаження на маршрутизатор і можуть порушити працездатність мережі;

– неправильно розміщені ACL-списки блокують дозволений трафік і дозволяють заборонений.

Типи і використання ACL-списків

Адміністратору доступно кілька варіантів створення списків контролю доступу. Складність вимог до структури визначає тип необхідного ACL-списку.

Існує три типи ACL-списків.

Стандартний ACL-список є найпростішим із трьох типів. При створенні стандартного ACL-списку для IP-протоколу, фільтрація по ACL-списках здійснюється на основі вихідної IP-адреси пакету. Адреса призначення пакету та порт не мають значення.

Стандартні ACL-списки визначають дозвіл пакетові на основі всього протоколу, такого як IP-протокол. Таким чином, при забороні вузлового пристрою стандартним ACL-списком, забороняються всі служби цього вузла. Такий тип ACL-списку корисний для дозволу доступу всіх служб визначеного користувача чи локальної мережі (LAN) через маршрутизатор із забороною доступу з інших IP-адрес.

Стандартні ACL-списки визначаються по номерах, які їм привласнюються. Номери з діапазону від 1 до 99 і від 1300 до 1999 привласнюються спискам доступу, що дозволяють або блокують IP-трафік.

Розширений ACL-список використовується для фільтрації не тільки по вихідній IP-адресі, а також по кінцевій IP-адресі, протоколу і номерах портів. Розширені ACL-списки використовуються частіше стандартних, оскільки вони є більш визначеними і забезпечують більш високий рівень контролю. Розширеним ACL-спискам привласнюються номери з діапазону від 100 до 199 і від 2000 до 2699.

Іменований ACL-список (NACL-список) має формат стандартного чи розширеного списку і позначається описовим ім'ям, а не номером. При налаштуванні іменованих ACL-списків, маршрутизатор IOS використовує режим підкоманди NACL.

Обробка ACL-списку

У списках контролю доступу міститься одна чи більше інструкцій. Кожна інструкція дає дозвіл, або забороняє трафік на основі зазначених параметрів. Трафік порівнюється з кожною інструкцією в ACL-списку одна за одною, поки не буде знайдене співпадіння, або не закінчиться список інструкцій.

Остання інструкція в ACL-списку завжди містить неявну заборону трафіку. Ця інструкція автоматично вставляється в кінець кожного ACL-списку, хоча і не є присутньою в ньому фізично. Неявна заборона блокує весь трафік. Ця можливість дозволяє запобігти випадковому проходженню небажаного трафіку.

Після створення списку контролю доступу, його необхідно застосувати до інтерфейсу, щоб задіяти його. ACL-список призначений для фільтрації вхідного або вихідного трафіку, що проходить через інтерфейс. Якщо пакет відповідає інструкції, що дозволяє проходження, то він пропускається маршрутизатором. Якщо ж пакет відповідає інструкції, що забороняє проходження, він зупиняється. ACL-список без жодної інструкції, що дозволяє проходження, приводить до блокування всього трафіку. Це пояснюється тим, що наприкінці кожного ACL-списку вказується неявна заборона. Таким чином, ACL-список буде перешкоджати проходженню всього трафіку, якщо не зазначені особливі дозволи.

Адміністратор може використовувати вхідний чи вихідний ACL-список для інтерфейсу маршрутизатора. Вхідний чи вихідний напрямок завжди розглядається з погляду маршрутизатора. Трафік, що надходить через інтерфейс, є вхідним, а трафік, який відправляється через інтерфейс – вихідним.

При отриманні пакету через інтерфейс маршрутизатор перевіряє такі параметри:

– наявність ACL-списку, пов'язаного з інтерфейсом;

– визначення типу ACL-списку (вхідний/вихідний);

– визначення відповідності трафіку умовам.

ACL-список, що використовується як вихідний до інтерфейсу, не діє для вхідного трафіку на тому ж інтерфейсі. Для кожного інтерфейсу маршрутизатор може мати один ACL-список для одного напрямку по кожному мережевому протоколу. Для IP-протоколу один інтерфейс може мати один вхідний і один вихідний ACL-список одночасно. ACL списки визначають набір правил, які дають додаткові можливості управління для пакетів, які надходять на вхідний інтерфейсу, пакетів, які передаються через маршрутизатор та пакетів, що виходять через вихідні інтерфейси маршрутизатора. ACL списки не діють на пакети, які виходять від самого маршрутизатора.

Вхідні ACL списки – вхідні пакети обробляються перед тим, як вони направляються на вихідний інтерфейс. Вхідні ACL є ефективними, оскільки економлять ресурси маршрутизатора, якщо пакет відкидається. Якщо пакет дозволений, потім він маршрутизується.

Вихідні ACL списки – вхідні пакети направляються на вихідний інтерфейс, а потім вони обробляються за допомогою вихідного ACL. Перед тим, як пакет направляється на вихідний інтерфейс, маршрутизатор перевіряє таблицю маршрутизації, щоб переконатися, що є маршрут для даного пакету. Якщо пакет не маршрутизується, він відкидається маршрутизатором. Потім маршрутизатор перевіряє, чи є ACL на вихідний інтерфейс. Для вихідних списків, «permit» означає послати пакет у вихідний буфер, а «deny» означає відкинути пакет.

ACL-списки, що застосовуються до інтерфейсу, створюють затримку трафіку. Навіть один довгий ACL-список може вплинути на продуктивність маршрутизатора .

Використання шаблонної маски

У простих ACL-списках вказується тільки одна дозволена чи заборонена адреса. Для блокування декількох адрес або діапазонів адрес необхідно кілька інструкцій або шаблонна маска. Використання IP-адреси мережі із шаблонною маскою забезпечує значно більшу гнучкість. За допомогою шаблонної маски можна блокувати діапазон адрес або всю мережу за допомогою лише однієї інструкції. У шаблонній масці використовуються символи «0» для вказівки частини IP-адреси, що повинна точно співпадати, і символи «1» – для частини IP-адреси, яка не повинна збігатися з визначеним номером. Шаблонна маска типу 0.0.0.0 вимагає точного співпадіння усіх 32 біт IP-адреси. Маска прирівнюється до використання параметра host. Шаблонна маска, яка використовується з функціями ACL-списків, аналогічна масці, що використовується в протоколі маршрутизації OSPF, але кожна маска має власну мету. При використанні з інструкціями ACL-списку шаблонна маска вказує вузол, або діапазон заборонених чи дозволених адрес. В інструкції ACL-списку IP-адреса та шаблонна маска утворюють поля, що порівнюються. Усі пакети, що входять або виходять через інтерфейс, порівнюються з кожною інструкцією ACL-списку для виявлення збігу. Шаблонна маска визначає, скільки біт вхідної IP-адреси відповідають порівнюваній адресі.

Наприклад: наступна інструкція дозволяє усі вузли мережі 192.168.1.0 і блокує інші:

access-list 1 permit 192.168.1.0 0.0.0.255

Шаблонна маска вказує, що повинні збігатися тільки перші три октети. Отже, якщо перші 24 біти вхідного пакету збігаються з першими 24 бітами порівнюваного поля, пакет дозволяється. Будь-який пакет з вихідною IP-адресою з діапазону 192.168.1.1 – 192.168.1.255 відповідає сполученню порівнюваної адреси і маски в зазначеному прикладі. Всі інші пакети забороняються ACL-списком за допомогою неявної інструкції deny any.

Оцінка результатів використання шаблонної маски

При створенні ACL-списку доступно два спеціальних параметри, які можна використовувати на місці шаблонної маски: host і any.

Параметр host

Для фільтрації одного визначеного вузла використовується шаблонна маска 0.0.0.0 після IP-адреси або параметр host перед IP-адресою.

R1(config)#access-list 9 deny 192.168.15.99 0.0.0.0

Що відповідає наступному:

R1(config)#access-list 9 deny host 192.168.15.99

Параметр any

Для фільтрації усіх вузлів використовуються всі параметри «1» шляхом налаштування шаблонної маски 255.255.255.255. При використанні шаблонної маски 255.255.255.255 вважається, що усі біти збігаються. Отже, IP-адреса, як правило, має вигляд 0.0.0.0. Іншим способом фільтрації усіх вузлів є використання параметра any.

R1(config)#access-list 9 permit 0.0.0.0 255.255.255.255

Що відповідає наступному:

R1(config)#access-list 9 permit any

Приклад, у якому забороняється визначений вузол і дозволяються всі інші:

R1(config)#access-list 9 deny host 192.168.15.99

R1(config)#access-list 9 permit any

Команда «permit any» дозволяє весь трафік, спеціально не заборонений ACL-списком. При такому налаштуванні, обробка пакетів не буде виконуватися до неявної команди deny any наприкінці ACL-списку.

У корпоративній мережі з ієрархічною схемою IP-адресації часто необхідна фільтрація трафіку підмережі.

Якщо 3 біти використовуються для розбивки мережі 192.168.77.0 на підмережі, маскою підмережі буде 255.255.255.224. У результаті вирахування маски підмережі з усіх значень маски 255 виходить шаблонна маска 0.0.0.31. Для дозволу вузлів у підмережі 192.168.77.32 використовується наступна інструкція ACL-списку:

access-list 44 permit 192.168.77.32 0.0.0.31

Перші 27 біт кожного пакету відповідають першим 27 бітам порівнюваної адреси. Загальний діапазон адрес, допустимих по цій інструкції, починається з 192.168.77.33 і закінчується 192.168.77.63. У нього входять всі адреси підмережі 192.168.77.32.

Створення правильних шаблонних масок для інструкцій ACL-списку забезпечує контроль, необхідний для точної оптимізації потоку трафіку.

Мережа 192.168.77.0 з маскою 255.255.255.192 чи /26 утворить наступні чотири підмережі:

192.168.77.0/26
192.168.77.64/26
192.168.77.128/26
192.168.77.192/26

Щоб створити ACL-список для фільтрації будь-яких з цих чотирьох підмереж, необхідно вирахувати маску підмережі 255.255.255.192 з усіх значень маски 255, у результаті чого вийде шаблонна маска 0.0.0.63. Щоб дозволити трафік з двох перших з цих підмереж, використовуються дві наступні інструкції ACL-списку:

access-list 55 permit 192.168.77.0 0.0.0.63
access-list 55 permit 192.168.77.64 0.0.0.63

Перші дві мережі в сумі утворять 192.168.77.0/25. У результаті вирахування підсумованої маски підмережі 255.255.255.128 зі значень маски 255 виходить шаблонна маска 0.0.0.127. Використання цієї маски дозволяє об’єднати ці дві підмережі в одній інструкції ACL-списку замість двох.

access-list 5 permit 192.168.77.0 0.0.0.127

Налаштування списків контролю доступу

Правильно складені списки контролю доступу позитивно позначаються на продуктивності та доступності мережі. Для досягнення максимальних результатів необхідне планування створення і розміщення списків контролю доступу.

Етап планування включає наступні дії:

1. Визначення вимог до фільтрації трафіку.

2. Вибір типу ACL-списку, який найкраще відповідає вимогам.

3. Визначення маршрутизатора та інтерфейсу, на якому буде застосовуватися ACL-список.

4. Вибір напрямку фільтрації трафіку.

Крок 1. Визначення вимог до фільтрації трафіку.

Список вимог до фільтрації трафіку, складається на основі опитування в кожному підрозділі підприємства. Ці вимоги різні для різних підприємств і ґрунтуються на потребах клієнтів, типах і об’ємах трафіку, а також задачах безпеки.

Крок 2. Вибір типу ACL-списку, що відповідає вимогам.

Вибір стандартного або розширеного ACL-списку обумовлений поточними вимогами до фільтрації. Вибір типу ACL-списку може вплинути на гнучкість фільтрації по ACL-списку, а також на продуктивність маршрутизатора і пропускну здатність мережі.

Стандартні ACL-списки легко створювати і впроваджувати. Однак фільтрація по стандартних ACL-списках можлива тільки на основі вихідної адреси і застосовується до всього трафіку без врахування його типу чи призначення. При маршрутизації в кілька мереж занадто близьке розміщення стандартного ACL-списку до джерела може ненавмисно

блокувати допустимий трафік. Отже, важливо розміщувати стандартні ACL-списки якнайближче до вузла призначення. У випадку більш складних вимог до фільтрації варто використовувати розширений ACL-список. Розширені ACL-списки дають більший контроль, ніж стандартні. Вони допускають фільтрацію по вихідних і кінцевих адресах. Ці списки також забезпечують фільтрацію по протоколу мережевого рівня, протоколу транспортного рівня і номерах портів, якщо це необхідно. Така, більш точна, фільтрація дозволяє адміністратору мережі створювати ACL-списки, що відповідають визначеним потребам плану по забезпеченню безпеки. Розширений ACL-список розміщується ближче до адреси джерела. Завдяки аналізу по вихідній і кінцевій адресі, ACL-список дозволяє блокувати пакети, що направляються у визначену кінцеву мережу перш, ніж вони залишать вихідний маршрутизатор. Пакети фільтруються перед тим, як вони перетнуть межі мережі, що допомагає підтримувати пропускну здатність.

Крок 3. Визначення маршрутизатора та інтерфейсу, для якого буде використовуватися ACL-список.

ACL-списки розміщуються на маршрутизаторах на рівні доступу або розподілу. Адміністратор мережі повинен мати контроль над цими маршрутизаторами і можливість реалізації політики безпеки. Без доступу до маршрутизатора адміністратор мережі не зможе налаштувати на ньому ACL-список. Вибір відповідного інтерфейсу залежить від вимог до фільтрації, типу ACL-списку і місця розташування призначеного маршрутизатора. Найкраще організовувати фільтрацію трафіку перш, ніж він досягне послідовного каналу з меншою пропускною здатністю.

Крок 4. Вибір напрямку фільтрації трафіку

При виборі напрямку, для якого буде використовуватися ACL-список, необхідно розглядати потік трафіку з погляду маршрутизатора. Вхідний трафік – це трафік, що надходить в інтерфейс маршрутизатора ззовні. Маршрутизатор порівнює вхідний пакет з ACL-списком перед пошуком мережі призначення в таблиці маршрутизації. Пакети, що відкидаються в цій точці, дозволяють виключити зайві операції пошуку маршрутизатора. Це робить вхідний список контролю доступу більш ефективним для маршрутизатора, ніж вихідний. Вихідний трафік проходить через маршрутизатор по інтерфейсу. Для вихідного пакету маршрутизатор уже здійснив пошук по таблиці маршрутизації і переключив пакет на правильний інтерфейс. Пакет порівнюється з ACL-списком безпосередньо перед виходом з маршрутизатора.

Налаштування ACL-списку

Після визначення вимог, планування списку контролю доступу і визначення розташування ACL-список необхідно налаштувати. Для кожного ACL-списку необхідний унікальний ідентифікатор. Ідентифікатор може бути числом або описовим ім’ям. У нумерованих списках контролю доступу, номер визначає тип створюваного ACL-списку:

– стандартним ACL-спискам для IP-протоколу привласнюються номери з діапазону від 1 до 99 і від 1300 до 1999;

– розширеним ACL-спискам для IP-протоколу привласнюються номери з діапазону від 100 до 199 і від 2000 до 2699.

Можна також створювати ACL-списки AppleTalk і IPX.

Обмеженням для будь-якого маршрутизатора є один ACL-список для протоколу і напрямку. Якщо на маршрутизаторі IP-протокол виконується в монопольному режимі, кожен інтерфейс може обробляти максимум два ACL-списки: один для вхідного й один для вихідного трафіку. Оскільки кожен ACL-список виконує порівняння кожного пакету, що проходить через підключення, використання ACL-списків створює затримку. Налаштування списку контролю доступу охоплює два етапи: створення і застосування.

Створення списку здійснюється в режимі глобальної конфігурації. За допомогою команди access-list вводяться інструкції списку контролю доступу. Поки список контролю доступу не буде готовий всі інструкції вводяться з однаковим номером ACL-списку.

Синтаксис стандартного ACL-списку наступний:

access-list [номер списку доступу] [deny|permit] [адреса джерела] [шаблонна маска джерела][log]

Оскільки кожен пакет порівнюється з інструкцією ACL-списку до знаходження співпадіння, порядок розміщення інструкцій у ACL-списку може впливати на створювану затримку. Тому інструкції потрібно розташовувати таким чином, щоб умови, які співпадають частіше в ACL-списку передували тим, які співпадають рідше. Наприклад, інструкції зі співпадінням по найбільшому об'єму трафіку необхідно розміщувати на початку ACL-списку. При цьому варто пам'ятати, що при співпадінні пакет більше не порівнюється з іншими інструкціями в ACL-списку. Це означає, що якщо один рядок дозволяє пакет, а наступний рядок у ACL-списку забороняє його, пакет буде дозволений. Тому варто планувати ACL-список таким чином, щоб інструкції з більш визначеними вимогами розташовувалися перед інструкціями з більш загальними вимогами. Іншими словами, забороняйте доступ визначеному вузлу в мережі, дозволяючи доступ іншим у всій мережі.

Для опису функції кожного розділу або окремої інструкції ACL-списку використовується команда remark:

access-list [номер списку] remark [текст]

Для видалення ACL-списку використовується команда:

no access-list [номер списку]

Зі стандартного або розширеного ACL-списку не можна видалити один рядок. ACL-список видаляється цілком і його необхідно замінити.

Фільтрація по ACL-списку неможлива до його застосування або призначення інтерфейсу.

ACL-список потрібно присвоїти одному або кільком інтерфейсам, вказавши вхідний чи вихідний трафік. Стандартний ACL-список потрібно використовувати якнайближче до адреси призначення.

R2(config-if)#ip access-group номер списку доступу [in | out]

Наступні команди дозволяють помістити список доступу access-list 5 для інтерфейсу Fa0/0 маршрутизатора R2 з фільтрацією вхідного трафіку:

R2(config)#interface fastethernet 0/0

R2(config-if)#ip access-group 5 in

За замовчуванням ACL-список на інтерфейсі застосовується на вихідний напрямок. Незважаючи на те, що вихідний напрямок встановлений за замовчуванням, дуже важливо вказувати напрямок для уникнення плутанини і для забезпечення фільтрації трафіку в правильному напрямку. Щоб видалити ACL-список з інтерфейсу без зміни самого ACL-списку, використовується команду no ip access-group інтерфейс. Деякі команди ACL-списку дозволяють оцінити правильність синтаксису, порядок інструкцій і розміщення на інтерфейсах:

– show ip interface – виводить інформацію про IP-інтерфейс та присвоєні ACL-списки;

– show access-lists [номер списку доступу] – дозволяє вивести вміст всіх ACL-списків маршрутизатора. Ця команда також виводить на екран число співпадінь по кожній інструкції з моменту застосування ACL-списку. Щоб вивести визначений список, потрібно додати ім'я ACL-списку або номер як параметр команди;

– show running-config – виводить на екран усі налаштовані ACL-списки маршрутизатора, навіть якщо вони в даний момент не застосовані до інтерфейсу.

При використанні нумерованих ACL-списків інструкції, що вводяться після створення ACL-списку, додаються в кінець. Такий порядок може не дати очікуваних результатів. Щоб вирішити цю проблему, потрібно видалити вихідний ACL-список та створити його заново.

Часто рекомендують створювати ACL-списки в текстовому редакторі. Це дозволить легко змінювати і вставляти ACL-список у конфігурацію маршрутизатора. Однак варто пам'ятати, що при копіюванні і вставці ACL-списку важливо спочатку видалити поточний ACL-список. В протилежному випадку всі інструкції будуть додані в кінець.

Налаштування нумерованих розширених ACL-списків

Розширені ACL-списки забезпечують більші можливості контролю в порівнянні зі стандартними. Розширені ACL-списки використовуються для дозволу або заборони трафіку по IP-адресі джерела, IP-адресі призначення, типу протоколу і номерах портів. Оскільки

розширені ACL-списки можуть бути точно визначеними, їхній розмір, як правило, швидко росте. Чим більше інструкцій містить ACL-список, тим складніше ним керувати.

Для розширених ACL-списків використовуються номери списку доступу з діапазонів від 100 до 199 і від 2000 до 2699. Правила, що діють для стандартних ACL-списків, також дійсні для розширених ACL-списків:

– в одному ACL-списку варто вказувати кілька інструкцій;

– кожна інструкція повинна мати той же номер ACL-списку;

– для представлення IP-адрес варто використовувати ключові слова host чи any.

Основною відмінністю синтаксису розширеного ACL-списку є необхідність вказувати протокол після умови дозволу або заборони. Це може бути IP-протокол із вказівкою всього IP-трафіку чи фільтрації визначеного IP-протоколу, такого як TCP, UDP, ICMP, OSPF.

Часто, поставлені вимоги можна виконати багатьма різними способами.

Наприклад, у компанії є сервер з адресою 192.168.3.75. Встановлено такі вимоги:

– дозволяти доступ до вузлів у локальній мережі 192.168.2.0;

– дозволяти доступ до вузла 192.168.1.66;

– блокувати доступ до вузлів у локальній мережі 192.168.4.0;

– дозволяти доступ до інших адрес у компанії.

Для задоволення цих вимог існують, як мінімум, два рішення. При плануванні ACL-списку потрібно намагатись зменшити кількість інструкцій, якщо це можливо.

Для зменшення числа інструкцій і скорочення навантаження на обробку маршрутизатором, можна використати наступні рекомендації:

– забезпечте виявлення прохідного трафіку великого об’єму і заборону трафіку, що блокується в початкових інструкціях ACL-списку, що дозволить уникнути порівняння пакетів з інструкціями далі за списком;

– об’єднуйте декілька інструкцій в одну інструкцію за допомогою діапазонів;

– намагайтеся блокувати доступ визначеної групи замість того, щоб дозволяти його іншій групі з більшою кількістю користувачів.

Налаштування іменних ACL-списків

Програмне забезпечення Cisco IOS версії 11.2 і вище дозволяє створювати іменовані ACL-списки (NACL-списки). У NACL-списку описове ім’я замінює числові діапазони, необхідні для стандартних і розширених ACL-списків. Іменовані ACL-списки мають можливості і переваги стандартних і розширених ACL-списків, при їхньому створенні відрізняється тільки синтаксис.

Ім’я ACL-списку є унікальним. Використання прописних букв в імені дозволяє зробити список легко впізнаваним у вихідних даних команд маршрутизатора і при рішенні проблем.

Для створення іменованого ACL-списку використовується команда:

ip access-list {standard | extended} ім’я

Після виконання цієї команди маршрутизатор переключається в режим підкоманди конфігурації NACL. Після вказівки початкової команди іменування необхідно ввести по одній всі інструкції. У NACL-списках використовується синтаксис команд стандартного, чи розширеного ACL-списку з інструкцією, що дозволяє або забороняє, на початку.

Іменований ACL-список застосовується до інтерфейсу аналогічно як застосовуються стандартні чи розширені ACL-списки.

При підключенні мережевого адміністратора до віддаленого маршрутизатора за допомогою протоколу Telnet на маршрутизаторі запускається вхідний сеанс. Telnet і SSH являють собою засоби внутрішньосмугового керування і використовують IP-протокол та мережеве підключення до маршрутизатора.

Метою обмеження доступу до віртуального терміналу (VTY) є підвищення рівня безпеки мережі. Зловмисники ззовні можуть спробувати отримати доступ до маршрутизатора. Якщо на порту віртуального маршрутизатора відсутній список контролю доступу, то кожен, хто зможе визначити ім’я користувача і пароль Telnet, отримає доступ до маршрутизатора. Якщо ACL-список застосувати до порту vty-маршрутизатора, по якому

дозволяється підключення тільки з визначених IP-адрес, то будь-якому користувачу, що намагається ввійти в маршрутизатор з IP-адреси, не дозволеного ACL-списком, буде відмовлено в доступі. При цьому варто пам’ятати, що можуть виникнути труднощі, якщо адміністратору доводиться підключатися до маршрутизатора з різних місць з різних IP-адрес.

Створення списку контролю доступу для віртуального терміналу здійснюється аналогічно списку для інтерфейсу. Однак для застосування ACL-списку до каналів VTY служить інша команда. Замість команди ip access-group використовується команда access-class.

При налаштуванні списків доступу для каналів віртуального терміналу, необхідно дотримуватись наступних рекомендацій:

– для каналів віртуального терміналу варто застосовувати не іменовані, а нумеровані ACL-списки;

– створювати однакові обмеження для всіх каналів віртуального терміналу, оскільки відсутня можливість контролювати канал, який використовує користувач.

Сеанси підключення до віртуального терміналу встановлюються між клієнтським ПЗ Telnet і кінцевим маршрутизатором. Мережевий адміністратор встановлює сеанс із кінцевим маршрутизатором, вводить ім’я користувача та пароль і вносить зміни в конфігурацію.

Тема 8. Бездротові мережі

Значення бездротових технологій у сучасному суспільстві. Порівняння дротових і бездротових мереж: переваги та обмеження. Основи бездротової передачі даних: радіохвилі та принципи їх поширення, спектри частот і методи модуляції, проблеми інтерференції та завадостійкості. Класифікація бездротових мереж: PAN (Personal Area Network): Bluetooth, ZigBee, WLAN (Wireless Local Area Network): Wi-Fi, MAN (Metropolitan Area Network): WiMAX, WWAN (Wireless Wide Area Network): стільникові мережі (3G, 4G, 5G, перспективи 6G).

Стандарти IEEE 802.11 (Wi-Fi): основні версії стандартів (802.11a/b/g/n/ac/ax), механізми доступу до середовища (CSMA/CA), архітектура Wi-Fi: точки доступу, клієнти, контролери. Мобільні мережі: принципи роботи стільникових мереж, архітектура 4G та 5G, особливості використання мобільного Інтернету для IoT. Бездротові технології для IoT: LoRaWAN, NB-IoT, ZigBee.

Безпека бездротових мереж: типові загрози (перехоплення, підробка точок доступу), методи захисту: WEP, WPA, WPA2, WPA3, використання VPN у бездротовому середовищі.

Значення бездротових технологій у сучасному суспільстві

Бездротові технології стали однією з ключових основ цифрової трансформації сучасного суспільства. Вони забезпечують мобільність користувачів, гнучкість розгортання інформаційно-комунікаційної інфраструктури та масштабованість сервісів. Саме бездротовий доступ зробив можливими такі явища, як масове використання смартфонів, розвиток хмарних сервісів, Інтернет речей (IoT), розумні міста, телемедицина, дистанційне навчання та віддалену роботу [11, 12].

У промисловості бездротові мережі використовуються для моніторингу технологічних процесів, збору телеметрії та побудови систем автоматизації. У транспорті вони забезпечують навігацію, керування потоками, взаємодію між транспортними засобами (V2V, V2X). У побуті бездротові технології інтегровані в розумні будинки, носимі пристрої та мультимедійні системи.

Таким чином, бездротові технології є фундаментом для формування цифрової економіки та інформаційного суспільства, де доступ до мережі та даних можливий у будь-який час і в будь-якому місці.

Порівняння дротових і бездротових мереж

Дротові та бездротові мережі мають різні характеристики, що визначають сфери їх ефективного застосування.

Дротові мережі традиційно забезпечують високу пропускну здатність, стабільність з’єднання та низькі затримки. Вони менш схильні до зовнішніх завад і простіші з точки зору забезпечення фізичної безпеки. Водночас їхніми обмеженнями є висока вартість прокладання кабелів, складність масштабування та відсутність мобільності.

Бездротові мережі, навпаки, характеризуються швидким розгортанням, гнучкістю та підтримкою мобільних користувачів. Вони не потребують фізичного підключення кожного пристрою, що особливо важливо для тимчасових або динамічних середовищ. Основними обмеженнями бездротових мереж є залежність якості зв’язку від умов середовища, обмеженість радіочастотного спектра, вплив інтерференції та підвищені вимоги до криптографічного захисту.

На практиці сучасні мережі здебільшого є гібридними, поєднуючи дротову магістральну інфраструктуру з бездротовими сегментами доступу.

Основи бездротової передачі даних

Бездротова передача даних ґрунтується на використанні електромагнітних хвиль, зокрема радіохвиль. Радіохвилі поширюються у просторі зі швидкістю світла та можуть відбиватися, заломлюватися, дифрагувати або поглинатися перешкодами. Якість зв’язку залежить від відстані між передавачем і приймачем, потужності сигналу, частоти, а також від характеристик навколишнього середовища [11, 12].

У реальних умовах поширення сигналу супроводжується багатопроменевістю, коли сигнал доходить до приймача кількома шляхами з різними затримками, що може спричиняти інтерференційні спотворення.

Радіочастотний спектр є обмеженим ресурсом і регулюється на міжнародному та національному рівнях. Для бездротових мереж використовуються як ліцензовані, так і неліцензовані діапазони. Наприклад, Wi-Fi працює переважно в неліцензованих діапазонах 2,4 ГГц, 5 ГГц та 6 ГГц.

Для передавання інформації застосовуються різні методи модуляції, зокрема амплітудна, частотна та фазова модуляція, а також їхні цифрові різновиди, такі як QPSK, QAM різних порядків. Сучасні системи використовують ортогональне частотне мультиплексування (OFDM), що дозволяє підвищити спектральну ефективність і завадостійкість.

Інтерференція виникає внаслідок накладання сигналів від різних джерел, що працюють в одному або суміжних частотних діапазонах. Для підвищення завадостійкості застосовуються методи кодування з виправленням помилок, адаптивний вибір модуляції та потужності, просторове різноманіття і технології MIMO.

Класифікація бездротових мереж

PAN – персональні мережі. Персональні бездротові мережі призначені для з'єднання пристроїв на невеликих відстанях. Технологія Bluetooth широко використовується для підключення гарнітур, периферійних пристроїв та носимої електроніки. ZigBee орієнтована на енергоефективні сенсорні мережі та автоматизацію, зокрема в системах «розумного дому».

WLAN – локальні бездротові мережі. WLAN, реалізовані на основі технології Wi-Fi, забезпечують бездротовий доступ до локальних і глобальних мереж у межах будівель або кампусів. Вони є основою корпоративних і домашніх мереж доступу.

MAN – міські мережі. MAN охоплюють значні території міста або регіону. Прикладом є технологія WiMAX, яка розроблялася для широкосмугового бездротового доступу на великих відстанях.

WWAN – глобальні бездротові мережі. WWAN базуються на стільникових технологіях. Мережі 3G забезпечили мобільний доступ до Інтернету, 4G (LTE) значно підвищили швидкість і зменшили затримки, а 5G орієнтовані на ультранизькі затримки, високу щільність підключень та підтримку IoT. Перспективи 6G пов'язані з використанням терагерцового діапазону, інтеграцією штучного інтелекту та підтримкою голографічних і тактильних сервісів.

Стандарти IEEE 802.11 (Wi-Fi)

Сімейство стандартів IEEE 802.11 визначає фізичний і канальний рівні WLAN. Початкові версії 802.11a/b/g забезпечували швидкості до десятків Мбіт/с. Стандарти 802.11n і 802.11ac запровадили MIMO та ширші канали, а 802.11ax (Wi-Fi 6) орієнтований на ефективну роботу в умовах високої щільності клієнтів.

Доступ до середовища реалізується за допомогою механізму CSMA/CA, який мінімізує колізії шляхом прослуховування каналу та використання випадкових затримок. Архітектура Wi-Fi мереж включає точки доступу, клієнтські пристрої та, у корпоративних середовищах, бездротові контролери для централізованого управління.

Мобільні мережі та IoT

Стільникові мережі базуються на поділі території на комірки, кожна з яких обслуговується базовою станцією. У 4G архітектура спрощена та орієнтована на пакетну передачу даних. 5G вводить сервісно-орієнтовану архітектуру, мережеву віртуалізацію та механізм network slicing.

Для IoT особливо важливими є технології з низьким енергоспоживанням і великим радіусом дії. До них належать LoRaWAN, NB-IoT та ZigBee, які дозволяють підключати тисячі сенсорів і пристроїв з мінімальними витратами енергії.

Бездротові мережі є більш уразливими до атак через відкритість середовища передавання. Типові загрози включають перехоплення трафіку, підробку точок доступу, атаки типу «людина посередині».

Для захисту використовуються криптографічні механізми. Ранні стандарти WEP виявилися ненадійними. WPA та WPA2 значно підвищили рівень безпеки, а WPA3 забезпечує захист від перебору паролів і покращене шифрування. Додатковим рівнем захисту є використання VPN, що забезпечує конфіденційність і цілісність даних у бездротовому середовищі.

Бездротові технології є невід'ємною складовою сучасних комп'ютерних мереж. Їх розвиток визначає можливості мобільного доступу, масштабування сервісів та впровадження нових цифрових рішень.

Тема 9. Глобальні мережі

Поняття глобальних мереж та їхня відмінність від локальних і регіональних мереж. Історичні передумови розвитку глобальних мереж (ARPANET, NSFNET, Internet). Архітектура та основні принципи побудови глобальних мереж. Базові компоненти: маршрутизатори, магістральні канали, точки обміну трафіком. Принцип багаторівневої ієрархії у структурі Інтернету. Провайдери та автономні системи. Технології глобальних мереж. Волоконно-оптичні лінії та підводні кабелі. Супутникові системи зв'язку. Мобільні технології (4G, 5G, перспективи 6G). Хмарні інфраструктури та розподілені обчислення як частина глобальних мереж. Протоколи та стандарти глобальних мереж. TCP/IP як універсальна основа. BGP (Border Gateway Protocol) і його роль у маршрутизації між автономними системами. DNS як глобальна служба імен. Системи управління адресним простором (IANA, ICANN, RIR). Забезпечення якості та безпеки у глобальних мережах. QoS (Quality of Service) у глобальних каналах. Загрози і виклики безпеці (DDoS-атаки, міжмережеві атаки, проблеми конфіденційності). Засоби захисту: VPN, шифрування, міжмережеві екрани, глобальні системи моніторингу. Концепція «Інтернету майбутнього». Інтернет речей (IoT) на глобальному рівні. Використання штучного інтелекту в управлінні глобальними мережами. Роль глобальних мереж у сучасному інформаційному суспільстві.

Глобальні комп'ютерні мережі є фундаментальною складовою сучасної інформаційної інфраструктури, що забезпечує об'єднання мільярдів пристроїв, користувачів і сервісів у єдиний комунікаційний простір. Вони суттєво відрізняються від локальних і регіональних мереж не лише за масштабами, а й за принципами побудови, управління, протоколами та соціально-економічною роллю.

Під глобальними мережами розуміють мережі зв'язку, які охоплюють значні географічні простори − країни, континенти або всю планету − і забезпечують передачу даних між віддаленими вузлами з використанням різноманітних телекомунікаційних технологій. На відміну від локальних мереж (LAN), що функціонують у межах будівлі або кампусу та перебувають під управлінням однієї організації, і регіональних або міських мереж (MAN), які обслуговують обмежену територію, глобальні мережі не мають єдиного централізованого власника чи адміністратора. Їхня робота ґрунтується на взаємодії багатьох незалежних операторів, провайдерів і автономних систем, об'єднаних спільними протоколами та правилами маршрутизації.

Історичні передумови розвитку глобальних мереж пов'язані з потребами наукових і військових структур у надійних системах обміну інформацією. Першим прообразом сучасного Інтернету стала мережа ARPANET, створена наприкінці 1960-х років у США в рамках проєктів Агентства передових оборонних досліджень (ARPA). Основною ідеєю ARPANET було впровадження пакетної комутації, що дозволяло зберігати працездатність мережі навіть у разі виходу з ладу окремих вузлів. У 1980-х роках подальший розвиток отримала мережа NSFNET, яка об'єднала університетські та дослідницькі центри і стала основною магістраллю для академічного Інтернету. Саме на базі цих ініціатив сформувався Internet − глобальна мережа мереж, що використовує єдиний стек протоколів TCP/IP і згодом стала доступною для комерційного та масового використання.

Архітектура глобальних мереж характеризується високим рівнем децентралізації та масштабованості. Її основу становить ієрархічна модель, у якій виділяють рівні доступу, агрегації та магістралі. На рівні доступу розташовані кінцеві користувачі та корпоративні мережі, які підключаються до провайдерів через різні технології − оптоволоконні, бездротові чи мобільні. Рівень агрегації об'єднує трафік від численних мереж доступу, оптимізує маршрути та забезпечує резервування каналів. Магістральний рівень формують високопродуктивні маршрутизатори та канали зв'язку надвисокої пропускної здатності, які передають дані між регіонами та континентами.

Ключовими компонентами глобальних мереж є маршрутизатори, що виконують інтелектуальну пересилку пакетів між мережами, магістральні канали зв'язку, здебільшого побудовані на основі волоконно-оптичних технологій, а також точки обміну інтернет-

трафіком (IXP). Останні відіграють особливо важливу роль, оскільки дозволяють провайдерам обмінюватися трафіком без залучення сторонніх магістральних операторів, зменшуючи затримки та фінансові витрати.

Структура Інтернету базується на концепції автономних систем − логічно та адміністративно незалежних мереж, кожна з яких має унікальний ідентифікатор ASN і власну політику маршрутизації. Автономні системи можуть належати інтернет-провайдерам, великим корпораціям, хмарним платформам або державним установам. Взаємодія між ними забезпечується спеціальними міждоменними протоколами, що дозволяють координувати маршрути передачі даних у глобальному масштабі.

Технологічною основою глобальних мереж є передусім волоконно-оптичні лінії зв'язку, які забезпечують надзвичайно високу пропускну здатність, низький рівень затримок і стійкість до електромагнітних завад. Особливе значення мають підводні оптичні кабелі, що з'єднують континенти та забезпечують основний обсяг міжконтинентального трафіку. Доповненням до них є супутникові системи зв'язку, які застосовуються для покриття віддалених або важкодоступних регіонів, а також у мобільних і резервних каналах передачі даних. У сучасних умовах важливу роль відіграють мобільні технології четвертого та п'ятого поколінь, які забезпечують високошвидкісний доступ до глобальних мереж з мінімальною затримкою. Перспективи розвитку 6G пов'язують із подальшим зростанням швидкостей, інтеграцією штучного інтелекту та підтримкою масових кіберфізичних систем.

Суттєвим елементом глобальних мереж стали хмарні інфраструктури та розподілені обчислення. Великі дата-центри, географічно розподілені по всьому світу, формують глобальні платформи зберігання та обробки даних, забезпечуючи доступність сервісів незалежно від місця перебування користувача. Хмарні сервіси тісно інтегровані з мережевою інфраструктурою та фактично є її логічним продовженням.

Функціонування глобальних мереж неможливе без уніфікованих протоколів і стандартів. Стек TCP/IP є універсальною основою Інтернету, забезпечуючи адресацію, маршрутизацію та надійну передачу даних. Особливе місце посідає протокол BGP, який використовується для обміну маршрутною інформацією між автономними системами та визначає шляхи поширення трафіку у глобальному масштабі. Не менш важливою є система доменних імен DNS, що реалізує розподілену ієрархічну службу перетворення символьних імен у IP-адреси. Управління адресним простором і доменними зонами здійснюється міжнародними організаціями IANA та ICANN у взаємодії з регіональними інтернет-реєстрами.

Одним із ключових викликів для глобальних мереж є забезпечення якості обслуговування та безпеки. QoS у глобальних каналах передбачає механізми пріоритизації трафіку, керування затримками, втратами пакетів і пропускною здатністю, що особливо важливо для мультимедійних та критично важливих сервісів. Водночас глобальні мережі є мішенню для численних загроз, зокрема розподілених атак відмови в обслуговуванні, міжмережевих атак і порушень конфіденційності даних. Для протидії цим загрозам застосовуються VPN-технології, криптографічні методи шифрування, міжмережеві екрани, системи виявлення вторгнень і глобальні платформи моніторингу мережевого трафіку.

Перспективи розвитку глобальних мереж пов'язують із концепцією «Інтернету майбутнього», що передбачає інтеграцію мільярдів пристроїв Інтернету речей на глобальному рівні, розвиток кіберфізичних систем і використання штучного інтелекту для автоматизованого управління мережами. Алгоритми машинного навчання вже сьогодні застосовуються для прогнозування перевантажень, оптимізації маршрутизації та виявлення аномалій у мережевому трафіку.

Таким чином, глобальні мережі відіграють ключову роль у сучасному інформаційному суспільстві, забезпечуючи основу для цифрової економіки, науки, освіти, державного управління та міжнародної комунікації. Їхній розвиток визначає не лише технічний прогрес, а й трансформацію соціальних процесів, формуючи нову глобальну інформаційну реальність.

Тема 10. Віртуалізація та автоматизація роботи мережі

Потреба у віртуалізації та автоматизації в умовах зростання складності мереж. Еволюція від традиційних до програмно-керованих мереж. Поняття віртуалізації мереж: віртуалізація як відокремлення апаратного та логічного рівнів, Network Functions Virtualization (NFV): заміна апаратних рішень віртуальними функціями, віртуалізація мережевих пристроїв: віртуальні маршрутизатори, комутатори, міжмережеві екрани. Технології віртуалізації мереж: віртуальні локальні мережі (VLAN) та оверлейні мережі (VXLAN), віртуальні приватні мережі (VPN) як форма віртуалізації каналів зв'язку, використання гіпервізорів та контейнерних платформ (VMware, KVM, Docker, Kubernetes). Програмно-конфігуровані мережі (SDN): архітектура SDN: контрольний і дата-площини, контролери SDN (OpenDaylight, Ryu, ONOS), протокол OpenFlow як основа для централізованого управління.

Потреба у віртуалізації та автоматизації в умовах зростання складності мереж

Сучасні комп'ютерні мережі зазнають стрімкого ускладнення внаслідок зростання обсягів переданих даних, кількості підключених пристроїв, поширення хмарних сервісів, мобільних технологій та Інтернету речей. Традиційні підходи до проєктування та адміністрування мереж, засновані на ручному налаштуванні фізичного обладнання, дедалі частіше виявляються недостатньо гнучкими, масштабованими та економічно ефективними. У таких умовах виникає об'єктивна потреба у впровадженні механізмів віртуалізації та автоматизації, які дозволяють зменшити залежність мережевих сервісів від апаратної інфраструктури, підвищити швидкість розгортання нових послуг і забезпечити централізоване управління мережею.

Автоматизація мережевих процесів зумовлена необхідністю мінімізації людського фактора, зменшення кількості помилок конфігурації та підвищення оперативності реагування на інциденти. Віртуалізація, у свою чергу, створює передумови для логічного поділу мережі на ізольовані сегменти, ефективного використання ресурсів та швидкого масштабування без фізичної модернізації обладнання.

Еволюція від традиційних до програмно-керованих мереж

Традиційні мережі будувалися за принципом тісної інтеграції апаратного та програмного забезпечення. Мережеві пристрої, такі як маршрутизатори й комутатори, поєднували в собі функції керування та пересилання даних, що ускладнювало централізоване управління та автоматизацію. Кожен пристрій конфігурувався окремо, що при великих масштабах мережі призводило до значних витрат часу та ресурсів.

Подальша еволюція мережевих технологій зумовила перехід до концепції програмно-керованих мереж, у яких логіка управління відокремлюється від процесів передачі даних. Це дозволило реалізувати нові підходи до адміністрування, засновані на централізованому контролі, політиках та програмному інтерфейсі управління мережею.

Поняття віртуалізації мереж

Віртуалізація мережі полягає у відокремленні логічного рівня мережевих функцій від фізичної апаратної інфраструктури. Такий підхід дає змогу створювати декілька логічно ізольованих мереж поверх спільних фізичних ресурсів. Кожна віртуальна мережа може мати власну топологію, адресний простір, правила маршрутизації та політики безпеки.

Ключовою ідеєю мережевої віртуалізації є абстрагування апаратних компонентів, що спрощує управління мережею та дозволяє швидко адаптувати її до змінних вимог бізнесу або навчального середовища.

Network Functions Virtualization (NFV)

Технологія Network Functions Virtualization спрямована на заміну традиційних апаратних мережевих рішень програмними реалізаціями, які виконуються у віртуальному середовищі. До таких функцій належать маршрутизація, міжмережеве екранування, балансування навантаження, системи виявлення вторгнень та інші мережеві сервіси.

NFV дозволяє розгортати мережеві функції у вигляді віртуальних машин або контейнерів на стандартному серверному обладнанні. Це значно знижує капітальні витрати, підвищує гнучкість масштабування та спрощує оновлення мережевих сервісів.

Віртуалізація мережевих пристроїв

Віртуалізація мережевих пристроїв передбачає створення програмних аналогів фізичних маршрутизаторів, комутаторів і міжмережевих екранів. Віртуальні маршрутизатори забезпечують маршрутизацію трафіку між логічними сегментами мережі, підтримуючи стандартні протоколи динамічної маршрутизації. Віртуальні комутатори виконують функції комутації на другому рівні моделі OSI та широко використовуються у хмарних і віртуалізованих середовищах. Віртуальні міжмережеві екрани дозволяють реалізувати політики безпеки без прив'язки до конкретного фізичного пристрою.

Технології віртуалізації мереж

Однією з базових технологій мережевої віртуалізації є віртуальні локальні мережі (VLAN), які дозволяють логічно сегментувати мережу на рівні канального шару незалежно від фізичного розташування пристроїв. VLAN забезпечують ізоляцію трафіку та підвищують рівень безпеки і керованості мережі.

Оверлейні мережі, зокрема VXLAN, розширюють можливості VLAN, дозволяючи створювати віртуальні мережі поверх IP-інфраструктури з підтримкою значно більшої кількості сегментів. VXLAN інкапсулює кадри другого рівня в пакети третього рівня, що робить можливим масштабування мереж у хмарних дата-центрах.

Віртуальна приватна мережа

Віртуальні приватні мережі VPN (скорочення від англ. virtual private network) є формою віртуалізації каналів зв'язку та забезпечують захищене передавання даних через публічні мережі. VPN дозволяють логічно об'єднувати віддалені сегменти мережі, створюючи ілюзію єдиного захищеного середовища.

Використання гіпервізорів і контейнерних платформ

Гіпервізори, такі як VMware ESXi або KVM, забезпечують виконання декількох віртуальних машин на одному фізичному сервері, що є основою для реалізації NFV та віртуальних мережевих пристроїв. Контейнерні платформи, зокрема Docker і Kubernetes, дозволяють розгортати мережеві сервіси у вигляді легковагових ізольованих контейнерів, що підвищує швидкість запуску та ефективність використання ресурсів.

Kubernetes відіграє важливу роль в автоматизації управління контейнеризованими мережевими функціями, забезпечуючи масштабування, балансування навантаження та відмовостійкість.

Програмно-конфігуровані мережі (SDN)

Програмно-конфігуровані мережі є логічним продовженням ідей віртуалізації та автоматизації. Основною особливістю SDN є розділення контрольної площини та площини передачі даних. Контрольна площина відповідає за прийняття рішень щодо маршрутизації та політик, тоді як дата-площина здійснює безпосередню пересилку пакетів.

Централізований SDN-контролер керує мережею через програмні інтерфейси, забезпечуючи глобальне бачення топології та стану мережі. До найпоширеніших контролерів належать OpenDaylight, Ryu та ONOS, які використовуються як у навчальних, так і в промислових середовищах.

Протокол OpenFlow

Протокол OpenFlow є одним з ключових елементів архітектури SDN і забезпечує взаємодію між SDN-контролером та мережевими пристроями. За допомогою OpenFlow контролер може динамічно змінювати правила обробки трафіку в комутаторах, визначати шляхи проходження пакетів та реалізовувати складні політики управління мережею.

Застосування OpenFlow дозволяє реалізувати централізоване, програмне та гнучке управління мережею, що є критично важливим для сучасних хмарних і масштабованих інфраструктур.

Тема 11
Віртуалізація та серверні середовища. Встановлення та основи адміністрування Windows Server

Концепція віртуалізації

Віртуалізація визначається як технологія, що може бути використана для створення віртуальних представлень серверів, сховищ, мереж та інших фізичних пристроїв. Віртуальне програмне забезпечення емулює функціональність фізичного обладнання для одночасного запуску віртуальних машин на одній фізичній машині. Підприємства використовують віртуалізацію для ефективного використання апаратних ресурсів та отримання додаткового прибутку від своїх інвестицій. Також ця технологія забезпечує надання послуг хмарних обчислень, що допомагають організаціям ефективно керувати своєю архітектурою.

Важливість віртуалізації зумовлена можливістю взаємодіяти з будь-яким апаратним ресурсом із більшою гнучкістю. Фізичні сервери споживають електроенергію, займають місце для зберігання та вимагають технічного обслуговування. Доступ до них часто обмежується фізичною близькістю та проектом мережі. Віртуалізація дозволяє усунути всі ці обмеження шляхом абстрагування функціональності фізичного обладнання у програмне забезпечення. Існує можливість здійснювати моніторинг, технічне обслуговування та використання апаратної інфраструктури як веб-додатка.

Для належного розуміння віртуальної машини на основі ядра (KVM) спочатку необхідно зрозуміти деякі базові поняття віртуалізації. Отож, віртуалізація – це процес, який дозволяє комп'ютеру спільно використовувати свої апаратні ресурси з кількома цифрове відокремленими середовищами. Кожне віртуалізоване середовище функціонує в межах виділених ресурсів, таких як пам'ять, обчислювальна потужність та сховище. Завдяки віртуалізації організації можуть перемикатися між різними операційними системами на одному сервері без перезавантаження [13].

Віртуальні машини та гіпервізори є двома важливими концепціями віртуалізації. Віртуальна машина визначається як програмно-визначений комп'ютер, що працює на фізичному комп'ютері з окремою операційною системою та обчислювальними ресурсами. Фізичний комп'ютер називається хост-машиною, а віртуальні машини називаються гостьовими машинами. На одній фізичній машині може працювати кілька віртуальних машин. Віртуальні машини абстрагуються від комп'ютерного обладнання за допомогою гіпервізора.

Огляд гіпервізорів

Гіпервізор – це програмний компонент, що здійснює керування декількома віртуальними машинами на комп'ютері. Цей механізм гарантує, що кожна віртуальна машина отримує виділені їй ресурси та не створює перешкод для функціонування інших віртуальних машин. Гіпервізор являє собою програмне забезпечення для віртуалізації, яке інсталюється на обчислювальні системи. Він виступає як програмний шар, що виконує роль посередника між віртуальними машинами та базовим апаратним забезпеченням або операційною системою хоста. Гіпервізори координують доступ до фізичного середовища таким чином, щоб декілька віртуальних машин мали доступ до власної частки фізичних ресурсів. Для прикладу, якщо віртуальна машина потребує обчислювальних ресурсів, таких як процесорна потужність комп'ютера, запит спочатку надсилається до гіпервізора. Після цього гіпервізор перенаправляє запит до базового апаратного забезпечення, яке виконує поставлене завдання. Існує два основних типи гіпервізорів [13].

Гіпервізори першого типу, також відомі як гіпервізори без операційної системи (також «рідні» гіпервізори) є програмами-гіпервізорами, які встановлюються безпосередньо на апаратне забезпечення комп'ютера, а не на операційну систему (рис. 2.1). Внаслідок такої архітектури гіпервізори першого типу характеризуються вищою. Гіпервізор першого типу зазвичай використовується для віртуалізації сервера. Наприклад, вони використовуються в центрах обробки даних та хмарних обчисленнях. Він працює безпосередньо на апаратному забезпеченні хоста і керує розподілом системних ресурсів між віртуальними операційними системами. Приклади гіпервізорів типу 1 включають VMware vSphere/ESxi, Xen і Oracle VM

Server.

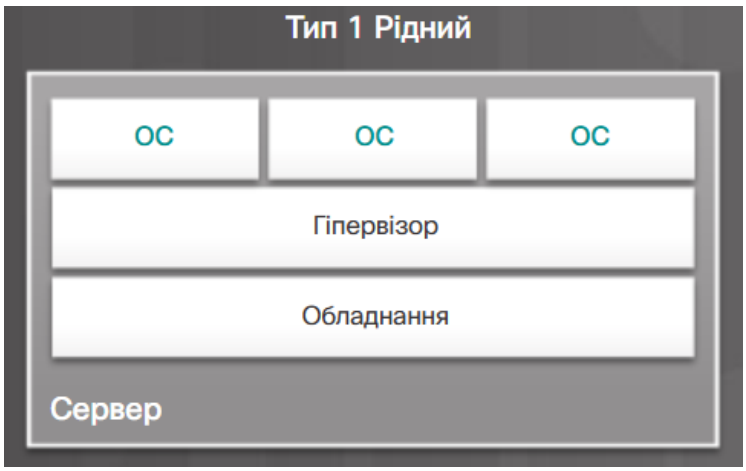

Рисунок 11.1 – Гіпервізор першого типу [14]

Гіпервізори другого типу функціонують як прикладне програмне забезпечення (додаток) на комп'ютерному обладнанні, що вже має встановлену операційну систему (рис. 11.2). Гіпервізори другого типу вважаються придатними для забезпечення обчислювальних потреб кінцевих користувачів. Гіпервізори другого типу, такі як VMware Workstation працюють з хост-комп'ютером для створення та використання кількох віртуальних машин. Гіпервізори типу 2 включають VMware Workstation, Windows Hyper-V, and Oracle VirtualBox.

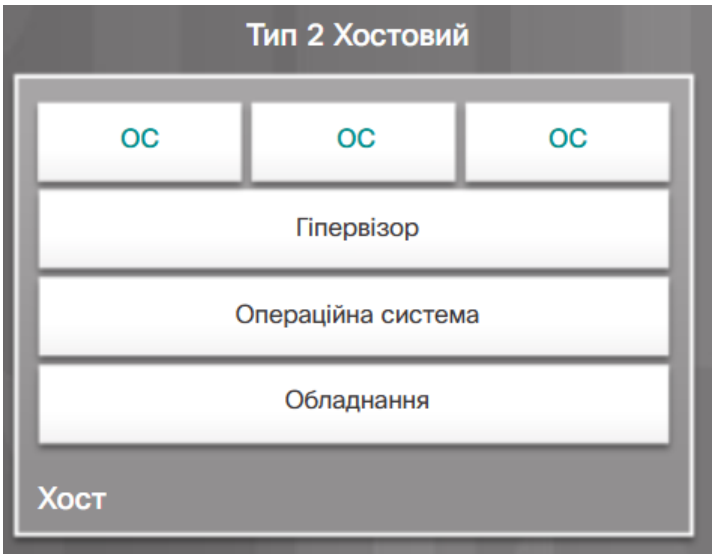

Рисунок 11.2 – Гіпервізор другого типу [14]

Створення віртуальних машин

Процес розгортання віртуалізації та створення віртаульних машин, з використанням гіпервізора другого типу Hyper-V, починається з активації відповідних компонентів операційної системи Windows. Для увімкнення Hyper-V у середовищі Windows використовуються такі інструменти, як PowerShell або інструмент обслуговування та керування образами розгортання (DISM). При використанні PowerShell необхідно ініціювати запуск оболонки з правами адміністратора, оскільки без цих привілеїв виконання команд буде неможливим. На робочому столі Windows здійснюється пошук Windows PowerShell через меню «Пуск», після чого через контекстне меню обирається опція «Запуск від імені адміністратора». Далі виконується команда Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V –All (рис. 11.3). Для завершення інсталяції та застосування змін необхідно ввести символ Y для перезавантаження комп'ютера [15].

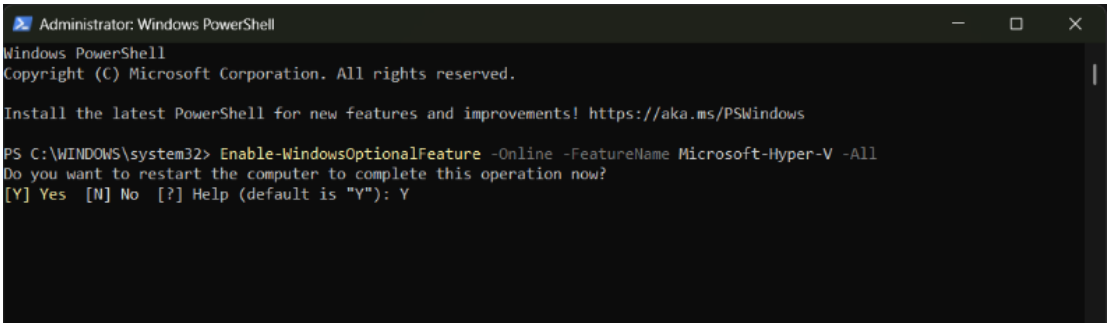

Рисунок 11.3 – Команда Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V –All [15]

Альтернативним методом є використання інструменту DISM, який дозволяє налаштовувати образи Windows та вмикати функції під час роботи операційної системи. Аналогічно до попереднього методу, запускається Windows PowerShell із правами адміністратора, після чого вводиться команда DISM /Online /Enable-Feature /All /FeatureName:Microsoft-Hyper-V. Успішне виконання операції підтверджується відповідним повідомленням у консолі.

Перед початком процесу створення віртуальної машини необхідно переконатися у відповідності системи певним вимогам. Комп’ютер повинен працювати під керуванням Windows Server або клієнтської версії Windows із вже активованою роллю Hyper-V. Користувач повинен мати членство в групі локальних адміністраторів або спеціалізованій групі «Адміністратори Hyper-V». Крім того, хост-система повинна мати достатній обсяг фізичної оперативної пам’яті для виділення віртуальній машині, а також достатній простір на дисковому накопичувачі для розміщення файлів конфігурації та віртуальних жорстких дисків. До необов’язкових, але рекомендованих умов належать наявність налаштованого віртуального комутатора для забезпечення мережевого підключення, а також наявність інсталяційного носія операційної системи (файлу .iso) або існуючого віртуального жорсткого диска (.vhd або .vhdx) [15].

Створення віртуальної машини може бути реалізовано за допомогою Диспетчера Hyper-V (Hyper-V Manager) або PowerShell. При використанні графічного інтерфейсу Диспетчера Hyper-V, який можна знайти через меню «Пуск», необхідно переконатися, що роль Hyper-V встановлена. У лівій області інтерфейсу обирається цільовий сервер, після чого на панелі «Дії» обирається пункт «Створити», а потім – «Віртуальна машина», що ініціює запуск «Майстра створення віртуальної машини».

На етапі «Вказати ім’я та розташування» вводиться ідентифікатор віртуальної машини та, за необхідності, змінюється місце зберігання файлів конфігурації шляхом встановлення прапорця «Зберігати віртуальну машину в іншому місці» та вибору відповідної папки (рис. 11.4).

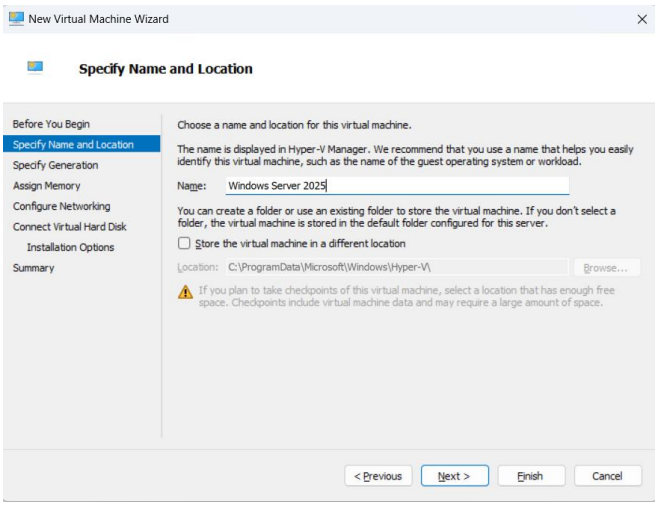

Рисунок 11.4 – Надання імені віртуальній машині та вибір місця зберігання [15]

Наступним кроком є вибір покоління віртуальної машини на сторінці «Вказати покоління». Рекомендується створення віртуальної машини покоління 2, якщо відсутні специфічні причини для використання покоління 1.

Процес конфігурації продовжується на сторінці «Призначити пам'ять», де визначається обсяг оперативної пам'яті, що виділяється при запуску. Існує можливість використання динамічної пам'яті, при цьому мінімальний обсяг становить 32 МБ, а максимальний обмежений ресурсами системи. На сторінці «Налаштування мережі» обирається віртуальний комутатор для інтеграції машини в мережу; цей крок можна пропустити та виконати налаштування пізніше.

На етапі «Підключення віртуального жорсткого диска» пропонуються варіанти: створення нового диска із зазначенням імені, розташування та розміру; використання існуючого диска (.vhd або .vhdx); або відкладення підключення диска на майбутнє (рис. 11.5).

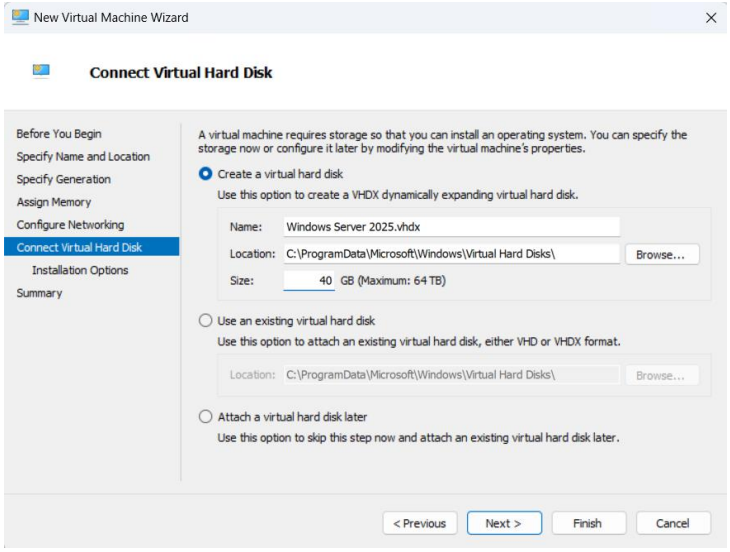

Рисунок 11.5 – Підключення віртуального жорсткого диска [15]

Завершальним етапом конфігурації є «Параметри інсталяції», де обирається джерело для встановлення операційної системи: інсталяція пізніше, використання файлу завантажувального образу (.iso), використання фізичного дисководу, або інсталяція з мережевого сервера. Після перевірки всіх параметрів на сторінці «Підсумок» процес завершується натисканням кнопки «Готово» (рис. 11.6).

Після завершення створення віртуальної машини виконується її запуск та підключення. У Диспетчері Hyper-V необхідно натиснути правою кнопкою миші на створену віртуальну машину та вибрати опцію «Підключитися...». У вікні підключення, що відкриється, для ініціалізації роботи системи обирається послідовність «Дія» та «Запустити».

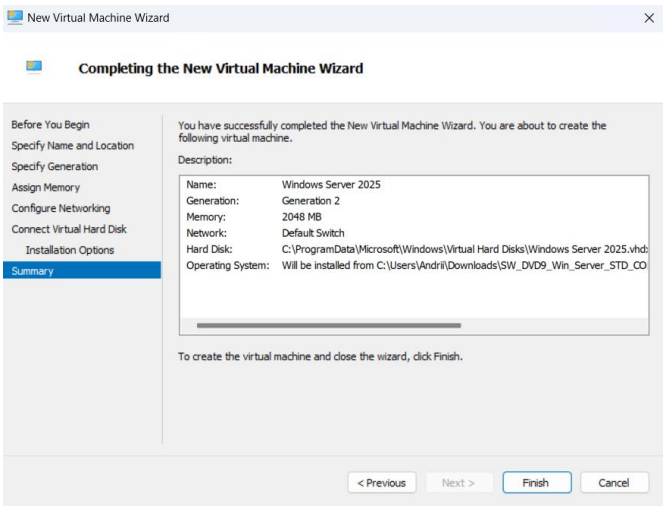

Рисунок 11.6 – Перевірка всіх параметрів на сторінці «Підсумок» [15]

Розуміння основ віртуалізації у Windows Server 2025

Віртуалізація у Windows Server 2025 визначається як трансформаційна технологія, що дозволяє створювати та експлуатувати декілька віртуальних машин (ВМ) на одному фізичному сервері або мережі взаємопов'язаних серверів, відомій як кластер. Кожна ВМ діє як незалежний комп'ютер, укомплектований власною операційною системою, додатками та виділеними ресурсами, функціонуючи ізольовано від інших ВМ. Ця технологія також поширюється на віртуалізацію пристроїв зберігання даних та мережевих ресурсів, що підвищує гнучкість та ефективність віртуалізованого середовища. Завдяки консолідації робочих навантажень на меншій кількості фізичних серверів, віртуалізація здатна суттєво зменшити витрати на обладнання, знизити енергоспоживання та мінімізувати фізичний простір, необхідний для дата-центрів [13].

Операційна система Windows Server 2025 включає Hyper-V – потужну функцію віртуалізації, яка забезпечує ефективне розгортання та керування ВМ як на клієнтських системах Windows, так і в серверних середовищах. Hyper-V, наступник раннього Windows Virtual PC, еволюціонував з моменту свого заснування у Windows Server 2008, ставши широко прийнятою та високо оціненою платформою серед системних адміністраторів. Його надійний набір служб та інструментів підтримує створення, конфігурацію та адміністрування ВМ, надаючи комплексне рішення для керування віртуальними середовищами [14].

Основа хмарної інфраструктури полягає в тому, що Hyper-V сприяє створенню та керуванню ВМ на фізичному сервері, забезпечуючи ефективне використання ресурсів. Ця здатність до віртуалізації є важливою для хмарних середовищ, де ресурси повинні динамічно розподілятися для задоволення змінних робочих навантажень. Опановуючи Hyper-V, IT-фахівці можуть використовувати цю технологію для створення масштабованих, гнучких хмарних інфраструктур.

Інтеграція з Microsoft Azure є ще одним ключовим аспектом, оскільки Microsoft Azure, одна з провідних хмарних платформ, значною мірою спирається на принципи віртуалізації, подібні до тих, що використовуються в Hyper-V. Розуміння Hyper-V дозволяє фахівцям безперешкодно переносити свої навички локальної віртуалізації в Azure, де вони можуть розгортати віртуальні машини Azure (Azure Virtual Machines) та використовувати такі функції, як Azure Site Recovery для аварійного відновлення. Ця обізнаність сприяє більш плавному процесу міграції та покращує загальні можливості керування хмарою [13].

Віртуалізація дозволяє забезпечити роботу декількох операційних систем (ОС) на одному фізичному сервері або в кластері серверів шляхом використання різних режимів. Кожен режим пропонує відмінні функції та переваги, адаптовані до різних потреб у віртуалізованих середовищах.

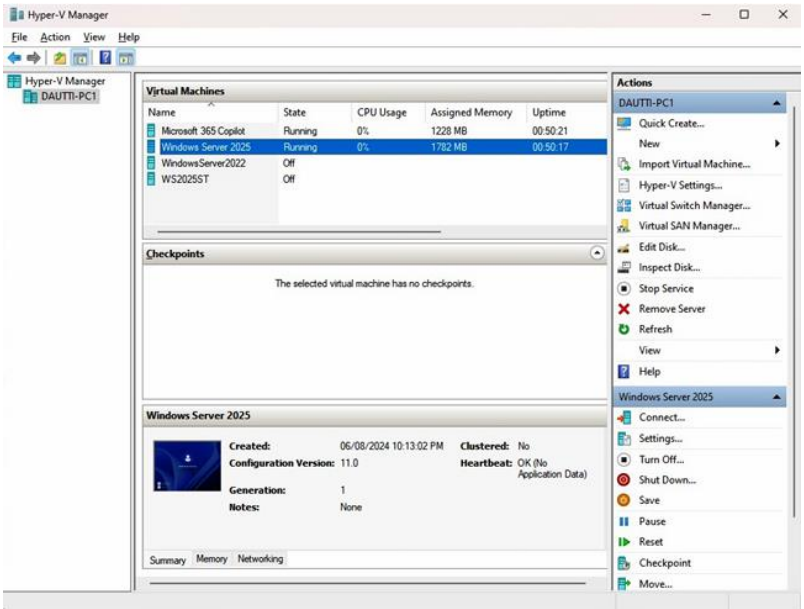

Рисунок 11.7 – Windows Server 2025, що працює в ізольованому та безпечному віртуальному середовищі [13]

Режим повної віртуалізації (Fully Virtualized Mode) передбачає, що кожна ОС працює у власному ізольованому та безпечному віртуальному середовищі, ніби вона знаходиться на окремій фізичній машині. Шар віртуалізації, або гіпервізор, керує ресурсами хост-сервера та розподіляє їх для кожної ВМ. Цей підхід забезпечує надійну ізоляцію між ВМ, гарантуючи, що вони можуть працювати незалежно, не змінюючи своїх конфігурацій. Повністю віртуалізовані середовища ідеально підходять для сценаріїв, що вимагають надійної безпеки та розділення, оскільки гостьові ОС залишаються необізнаними про базову інфраструктуру віртуалізації. Рисунок 11.7 ілюструє, як Windows Server 2025 функціонує в такому ізольованому середовищі, підкреслюючи розділення ресурсів та процесів між різними ВМ.

Режим паравіртуалізації (Paravirtualized Mode) пропонує більш інтегрований підхід, дозволяючи гостьовій ОС безпосередньо спілкуватися з гіпервізором. На відміну від режиму повної віртуалізації, де гостьова ОС не знає про шар віртуалізації, паравіртуалізовані системи вимагають модифікації гостьової ОС для ефективної взаємодії з гіпервізором. Цей режим використовує інтерфейс прикладного програмування (API) для забезпечення прямого зв’язку, що зменшує накладні витрати, зазвичай пов’язані з емуляцією обладнання. Результатом є значне підвищення продуктивності та використання ресурсів, що робить його переконливим вибором для середовищ, де ефективність та швидкість мають першорядне значення [13].

Режим контейнеризації (Containerization Mode) фокусується на інкапсуляції додатків разом з їхніми середовищами виконання, системними інструментами та налаштуваннями в самодостатні одиниці, відомі як контейнери. На відміну від ВМ, які віртуалізують цілі операційні системи, контейнери віртуалізують на рівні додатків, забезпечуючи легке та портативне рішення. Кожен контейнер працює незалежно, але використовує спільне ядро хост-ОС, що робить його ефективним вибором для розгортання та керування додатками в різних середовищах. Контейнери підвищують масштабованість, оптимізують розгортання додатків та забезпечують узгодженість, упаковуючи всі необхідні компоненти разом. Цей підхід є особливо корисним для розробки, тестування та послідовного розгортання додатків на різних платформах [13].

Варто зазначити, що фізичний сервер працює з операційною системою, відомою як хост-ОС, яка керує апаратними ресурсами сервера. Натомість, ВМ запускає операційну систему, що називається гостьовою ОС, яка працює у віртуальному середовищі, створеному хост-ОС або гіпервізором. Наприклад, у певній конфігурації ноутбук може бути обладнаний Windows 11 Pro як хост-ОС, тоді як ВМ на тій самій машині запускає Windows Server 2025 Standard як гостьову ОС. Хост-ОС відіграє критичну роль у розподілі та контролі апаратних ресурсів, дозволяючи гостьовій ОС безперешкодно функціонувати у віртуалізованому середовищі.

У віртуалізації продуктивність є критичним фактором, який може суттєво впливати на ефективність розгорнутих ВМ. Розуміння впливу різних типів сховищ та конфігурацій інфраструктури є важливим для оптимізації продуктивності ВМ.

При проектуванні віртуалізованого середовища розуміння відмінностей між мережами зберігання даних (SAN) та локальними дисками є вирішальним, оскільки кожне рішення для зберігання представляє унікальні характеристики продуктивності та наслідки для операцій ВМ.

Мережі зберігання даних (SAN) забезпечують централізоване рішення для зберігання, яке дозволяє декільком серверам отримувати доступ до спільного пулу ресурсів зберігання. Хоча SAN пропонують такі переваги, як висока доступність та масштабованість, вони можуть вносити затримку, що впливає на продуктивність. Ця затримка виникає через мережеві накладні витрати, оскільки дані повинні проходити через мережеву фабрику, щоб досягти масиву зберігання. У середовищах з високим попитом, де швидкий доступ до даних є критичним, ця затримка може призвести до повільнішої продуктивності ВМ, особливо для додатків з інтенсивним вводом-виводом.

Натомість, локальні диски безпосередньо підключені до фізичного сервера, що розміщує ВМ. Ця конфігурація зазвичай забезпечує меншу затримку, оскільки даним не потрібно проходити через мережу. Локальне сховище ідеально підходить для додатків, що вимагають швидкого доступу до даних та високої пропускної здатності, таких як бази даних

та системи обробки транзакцій. Однак локальні диски обмежують надлишковість та масштабованість порівняно з SAN, що робить їх менш придатними для великих віртуалізованих середовищ.

Гіперконвергентна інфраструктура (HCI) інтегрує обчислення, зберігання та мережі в єдине програмно-кероване рішення, пропонуючи уніфікований підхід до віртуалізації. HCI може підвищити продуктивність кількома способами. По-перше, покращена локальність даних досягається шляхом використання локального сховища в кожному вузлі кластера HCI, що значно скорочує час доступу до даних. Ця локальність мінімізує затримку, забезпечуючи швидшу продуктивність ВМ, особливо для додатків з високими вимогами до вводу-виводу. По-друге, масштабованість та еластичність дозволяють організаціям безперешкодно масштабувати свої ресурси. Зі зростанням попиту до кластера можна додавати додаткові вузли, ефективно розподіляючи робочі навантаження та підвищуючи продуктивність без шкоди для цілісності системи зберігання. По-третє, інтелектуальне керування ресурсами реалізується у багатьох рішеннях HCI, які включають передову аналітику та функції керування ресурсами, що оптимізують розміщення робочих навантажень на основі метрик продуктивності. Ця здатність гарантує, що ВМ розподіляються на найбільш відповідні ресурси, додатково підвищуючи загальну продуктивність [13].

Вибір типу сховища – чи то SAN, чи локальні диски – безпосередньо впливає на продуктивність віртуалізованих середовищ. Крім того, використання HCI може надати значні переваги в ефективному керуванні ресурсами та забезпеченні оптимальної продуктивності для ВМ. Розуміння цих режимів віртуалізації дозволяє обрати найбільш відповідний метод для інфраструктури залежно від вимог до продуктивності, безпеки та масштабованості.

Додавання та налаштування ролі Hyper-V у Windows Server 2025

Значною перевагою серверної віртуалізації є здатність запускати декілька віртуальних машин на одному фізичному сервері, максимізуючи при цьому продуктивність та ефективність використання ресурсів. Hyper-V дозволяє безперешкодно створювати, керувати та експлуатувати ВМ у середовищі Windows Server 2025.

Для повного розуміння архітектури Hyper-V доцільно представити її як деревоподібну структуру, де гіпервізор діє як корінь і глибоко інтегрований у фундамент апаратного забезпечення, що слугує ґрунтом. Гіпервізор є фундаментальним компонентом віртуальної платформи Hyper-V, маючи прямий доступ до апаратних ресурсів фізичного сервера, включаючи центральний процесор, пам’ять, сховище та мережеві ресурси. Цей прямий контроль дозволяє гіпервізору ефективно керувати цими ресурсами та розподіляти їх між декількома ВМ [13].

З гіпервізора розгалужуються окремі середовища виконання, відомі як розділи. Кожен розділ є ізольованим, що означає його незалежне функціонування без втручання з боку інших. Така ізоляція є критично важливою для безпеки та стабільності, оскільки вона гарантує, що проблема в одному розділі не вплине на інші. Самі розділи не мають прямого доступу до фізичного обладнання; натомість вони взаємодіють із віртуалізованим шаром, що надається гіпервізором або кореневим розділом. Цей віртуалізований шар абстрагує апаратне забезпечення, надаючи гостьовим операційним системам узгоджене та кероване середовище для роботи.

Кореневий розділ (root) є першим і найбільш привілейованим розділом, у якому працюють як хост-операційна система, так і роль Hyper-V. Він діє як центральний вузол, керуючи взаємодією з апаратним забезпеченням та наглядаючи за створенням і роботою інших розділів, відомих як дочірні розділи. Ці дочірні розділи розміщують гостьові операційні системи, які можуть включати різні версії Windows або Linux, забезпечуючи різноманітне та гнучке обчислювальне середовище. Зв’язок між кореневим і дочірніми розділами забезпечується спеціалізованими компонентами, відомими як Постачальник послуг віртуалізації (Virtualization Service Provider – VSP) та Споживач послуг віртуалізації (Virtualization Service Consumer – VSC), як проілюстровано на рисунку 11.8.

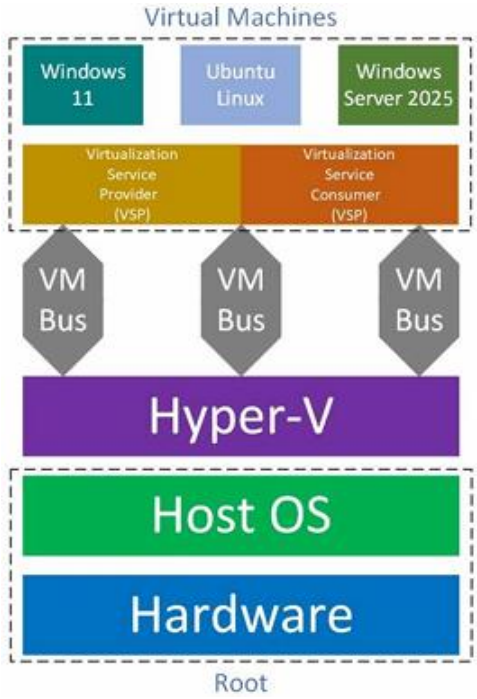

Рисунок 11.8 – Архітектура Hyper-V [13]

Ці компоненти використовують логічний канал зв'язку, що називається Шиною віртуальної машини (Virtual Machine Bus – VMBus), для ефективного обміну даними та командами. Така схема гарантує, що гостьові операційні системи в дочірніх розділах можуть виконувати необхідні операції, такі як доступ до сховища або мережевих ресурсів, без прямого доступу до апаратного забезпечення. Розуміння цієї архітектури є ключовим для оцінки того, як Hyper-V підтримує різні типи віртуалізації, включаючи повну віртуалізацію, де гостьова операційна система працює без змін у віртуальному середовищі. Ця можливість є критичною для сценаріїв, що вимагають сумісності та мінімальної модифікації існуючого програмного забезпечення. Однак перед розгортанням Hyper-V необхідно ознайомитися зі специфічними апаратними та програмними вимогами, а також передумовами для забезпечення успішної реалізації та оптимальної продуктивності [13].

Перед інсталяцією та використанням Hyper-V критично важливо переконатися, що сервер відповідає специфічним передумовам, необхідним для активації гіпервізора. Основною вимогою є підтримка сервером віртуалізації на апаратному рівні, що є основою функціональності Hyper-V. Це передбачає наявність процесора з увімкненою технологією віртуалізації, такою як Intel VT-x або AMD-V [13].

Ці технології є важливими, оскільки дозволяють процесору ефективно керувати декількома ВМ, динамічно та безпечно розподіляючи ресурси без значних накладних витрат. Окрім підтримки віртуалізації, сервер також повинен мати увімкнені інші функції, такі як запобігання виконанню даних (Data Execution Prevention – DEP), що забезпечує додатковий рівень безпеки, запобігаючи виконанню шкідливого коду в захищених областях пам'яті. Крім того, прошивка BIOS або UEFI сервера повинна бути налаштована коректно для підтримки цих технологій, з увімкненими опціями віртуалізації.

Ще одним важливим фактором є вкладена віртуалізація, особливо в середовищах, де планується запуск Hyper-V всередині ВМ. Ця розширена функція дозволяє створювати віртуальні середовища всередині ВМ, забезпечуючи гнучкість для тестування, розробки та навчальних сценаріїв без потреби в додатковому фізичному обладнанні. Окрім цих апаратних вимог, також важливо переконатися, що операційна система є сумісною з Hyper-V. Наприклад, Hyper-V доступний лише в певних редакціях Windows Server та клієнтських операційних систем Windows, таких як Windows 11 Pro або Enterprise. Забезпечення відповідності сервера цим умовам є ключовим для досягнення плавного та ефективного розгортання Hyper-V, що дозволяє повною мірою використовувати переваги віртуалізації в IT-середовищі [13].

Вкладена віртуалізація визначається як розширена функція, що дозволяє запускати ВМ всередині іншої ВМ. По суті, це означає, що апаратне забезпечення хост-машини здатне запускати Hyper-V всередині гостьової операційної системи, дозволяючи гостьовій ОС створювати та керувати додатковими ВМ так само, якби вона працювала безпосередньо на фізичному обладнанні. Ця можливість, хоча спочатку може здаватися теоретичною, підтримується компанією Microsoft починаючи з Windows Server 2016 і стала неоціненним інструментом у різних сценаріях, таких як тестування складних конфігурацій, навчальні середовища або запуск віртуальних лабораторій, де потрібні кілька рівнів віртуалізації. Простіше кажучи, вкладена віртуалізація дозволяє розглядати гостьову операційну систему так, ніби вона є хост-операційною системою, запускаючи Hyper-V і створюючи віртуалізоване середовище всередині вже віртуалізованого середовища. Таке вкладене налаштування може бути особливо корисним для сценаріїв, що вимагають декількох ізольованих середовищ, або для розробників та IT-фахівців, яким необхідно тестувати розгортання та конфігурації безпечним і контрольованим чином.

Налаштування вкладеної віртуалізації у Windows Server 2025 виконується безпосередньо за допомогою Windows PowerShell. Спочатку через контекстне меню кнопки «Пуск» обирається Windows PowerShell (з правами адмінічтратора). У вікні PowerShell виконується така команда: Set-VMProcessor -VMName <VMName> -ExposeVirtualizationExtensions $true. Ця команда дозволяє гостьовій ВМ надавати необхідні розширення віртуалізації, дозволяючи їй запускати Hyper-V. Наступна команда: Get-VMNetworkAdapter -VMName <VMName> | Set-VMNetworkAdapter -MacAddressSpoofing On вмикає підміну MAC-адреси (MAC address spoofing) на мережевому адаптері ВМ, що є необхідним для функціонування мережі в сценарії вкладеної віртуалізації. Після виконання цих конфігурацій можна переходити до інсталяції Hyper-V всередині гостьової ВМ, дотримуючись інструкцій, які розглядають інсталяцію Hyper-V на Windows Server 2025 [13].

Ознайомлення з диспетчером Hyper-V для адміністрування віртуальних машин

Диспетчер Hyper-V (Hyper-V Manager) – це універсальний та важливий інструмент для адміністрування віртуальних машин у середовищі Windows Server 2025. Він забезпечує централізований інтерфейс для керування різноманітними завданнями, пов’язаними з ВМ, оптимізуючи адміністрування віртуалізованих ресурсів [13].

За допомогою Диспетчера Hyper-V виконується ефективне створення нових ВМ, імпорт існуючих та видалення тих, що більше не потрібні, що забезпечує гнучкість у керуванні віртуальною інфраструктурою. Інструмент також дозволяє налаштовувати та керувати віртуальними комутаторами, які є критично важливими для підключення ВМ до мережі та забезпечення їх ефективної взаємодії з іншими мережевими ресурсами. Додатково Диспетчер Hyper-V сприяє створенню менеджера мережі зберігання даних (SAN), що дозволяє ВМ підключатися до рішень спільного зберігання. Ця можливість є життєво важливою для підтримання високої доступності та продуктивності у віртуальному середовищі. Крім того, Диспетчер Hyper-V включає функції для інспекції та оптимізації віртуальних дисків, дозволяючи регулювати розподіл дискового простору та покращувати продуктивність відповідно до вимог. Також здійснюється керування станом ВМ шляхом їх зупинки або вимкнення, що є корисним для цілей обслуговування та усунення несправностей.

Інтерфейс Диспетчера Hyper-V у Windows Server 2025 організований у п’ять основних секцій: панель серверів, що відображає список серверів; панель ВМ, що показує ВМ на вибраному сервері; панель контрольних точок, яка надає доступ до контрольних точок ВМ для відновлення станів; деталі вибраної ВМ, що пропонують інформацію та налаштування для поточної вибраної ВМ; та панель дій, яка надає доступ до різних управлінських дій (рис. 11.9). Кожен із цих компонентів відіграє вирішальну роль у ефективному керуванні та конфігурації віртуального середовища.

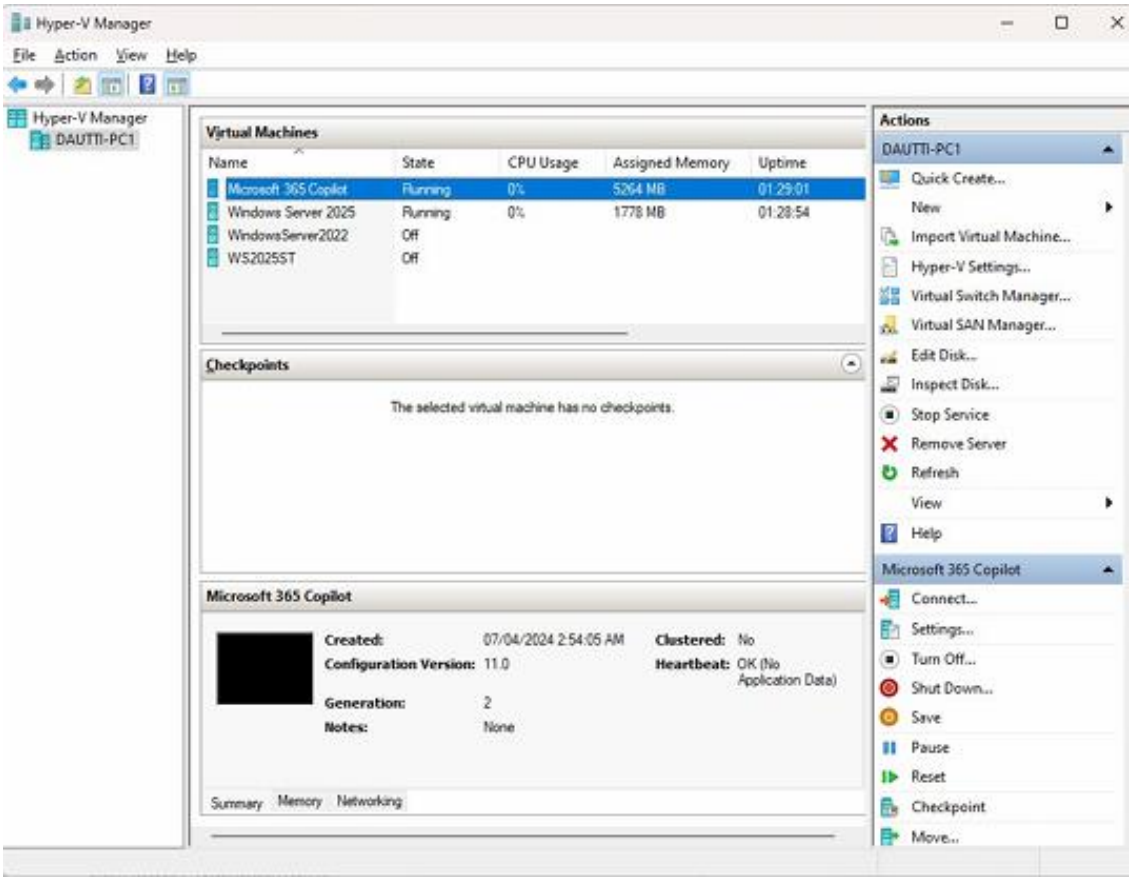

Рисунок 11.9 – Інтерфейс Диспетчера Hyper-V у Windows Server 2025 (основні панелі) [13]

Під час навігації Диспетчером Hyper-V необхідно ознайомитися зі специфічними елементами інтерфейсу користувача, які відіграють критичну роль у керуванні віртуальними середовищами. Колонка «Стан реплікації» (Replication Health) у вікні віртуальних машин надає цінну інформацію про статус реплікації ВМ. Ця функція є вирішальною для підтримки цілісності даних та доступності, особливо у сценаріях аварійного відновлення. Реплікація дозволяє дублювати ВМ на вторинний хост, гарантуючи наявність резервної копії у разі збою. Статус реплікації може відображати такі індикатори, як «Нормальний», «Попередження» або «Критичний», кожен з яких представляє поточний стан процесу реплікації. Регулярний моніторинг стану реплікації допомагає проактивно виявляти та вирішувати потенційні проблеми; статус «Нормальний» вказує на коректне функціонування, тоді як попередження або критичні сповіщення вимагають негайного розслідування для захисту доступності даних у ситуаціях відмови [13].

Іншою важливою функцією, доступною в Диспетчері Hyper-V, є можливість експорту віртуальної машини на інший хост. Ця функція є необхідною для керування ресурсами та операційної безперервності. Експорт ВМ передбачає створення повної копії, включаючи налаштування конфігурації, віртуальні жорсткі диски та будь-які пов’язані знімки. Ця функціональність дозволяє безперешкодно мігрувати ВМ між різними хостами Hyper-V. Розуміння процесу експорту ВМ озброює IT-фахівців навичками, необхідними для адаптації віртуальних середовищ до змінних потреб організації, забезпечуючи мінімальні збої під час обслуговування або балансування робочих навантажень між хостами.

Перед початком створення та керування віртуальними машинами необхідно ефективно налаштувати параметри Hyper-V на сервері. Ці налаштування доступні через опцію «Параметри Hyper-V...» (Hyper-V Settings...), що знаходиться на панелі дій Диспетчера Hyper-V (рис. 11.10). Кілька ключових областей конфігурації можуть бути адаптовані для оптимізації віртуального середовища, як показано нижче.

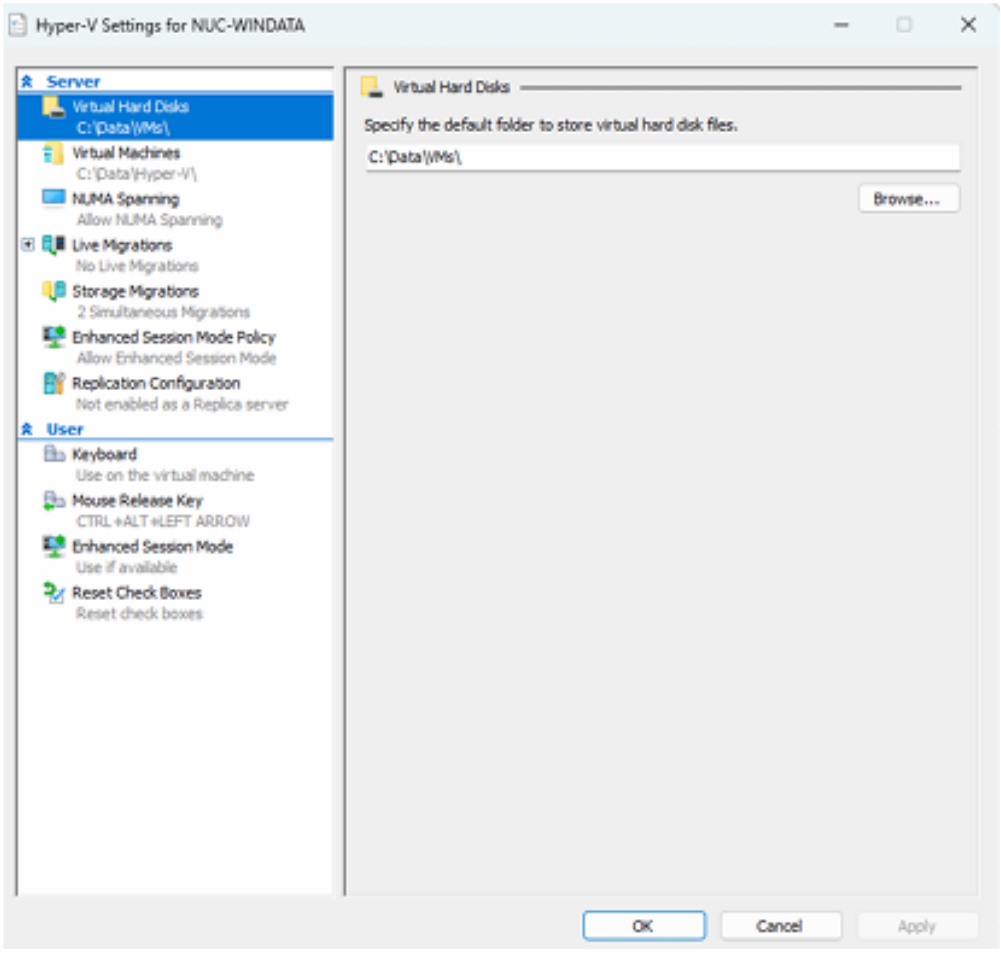

Рисунок 11.10 – Параметри Hyper-V (Hyper-V Settings) у Windows Server 2025 [13]

Налаштування «Віртуальні жорсткі диски» дозволяє вказати каталог за замовчуванням для зберігання файлів віртуальних дисків; належна конфігурація тут є життєво важливою для підтримки організованого зберігання та ефективного використання дискового простору. Опція «Віртуальні машини» дозволяє визначити розташування за замовчуванням для файлів конфігурації ВМ, забезпечуючи централізоване керування всіма налаштуваннями та метаданими. Розділ «Фізичні графічні процесори» (Physical GPUs) дозволяє призначити графічний процесор, який буде використовуватися ВМ, що є особливо важливим для робочих навантажень, які вимагають високої графічної продуктивності. Налаштування «NUMA Spanning» контролює, чи можуть ВМ охоплювати кілька вузлів неоднорідного доступу до пам'яті (NUMA), що може підвищити продуктивність ВМ шляхом доступу до ширшого діапазону обчислювальних ресурсів. Конфігурація «Міграції сховища» дозволяє встановити максимальну кількість одночасних міграцій сховища, які може обробляти сервер, що є критичним для підтримки продуктивності під час переміщення даних. Політика розширеного режиму сеансу (Enhanced Session Mode Policy) вмикає або вимикає можливість перенаправлення локальних пристроїв та ресурсів з хост-машини до підключення віртуальної машини. Додатково важливо відзначити функції конфігурації реплікації (Replication Configuration) для налаштування Hyper-V Replica та опцію живих міграцій (Live Migrations), що полегшує безперешкодне переміщення запущених ВМ без простою.

Для створення та керування віртуальним жорстким диском (VHD) на Windows Server 2025 за допомогою Диспетчера Hyper-V виконується певна послідовність дій. Спочатку запускається Диспетчер Hyper-V через меню «Засоби Windows». На панелі дій обирається «Створити» (New), а потім «Жорсткий диск...», що ініціює запуск Майстра створення нового віртуального жорсткого диска (рис. 11.11).

Рисунок 11.11 – Процес створення віртуального жорсткого диска [13]

У майстрі обирається формат VHD: VHDX для підвищеної продуктивності та більшої ємності або VHD для сумісності зі старими системами. Далі визначається тип диска: фіксованого розміру (розмір встановлюється і залишається постійним), динамічно розширюваний (розмір зростає в міру додавання даних до максимуму) або диференціальний (відстежує зміни від базового VHD). Вказується ім’я та розташування файлу VHD, після чого приймається рішення про створення порожнього диска або імпорт даних з існуючого фізичного диска. Після перевірки вибору натискається кнопка «Готово» для створення VHD.

Налаштування розподілу оперативної пам’яті для ВМ є критичним для оптимізації продуктивності. Процес починається з переконання, що ВМ вимкнена. У Диспетчері Hyper-V натискається права кнопка миші на вибраній ВМ і обирається пункт «Параметри...» (Settings...) (рис. 11.12).

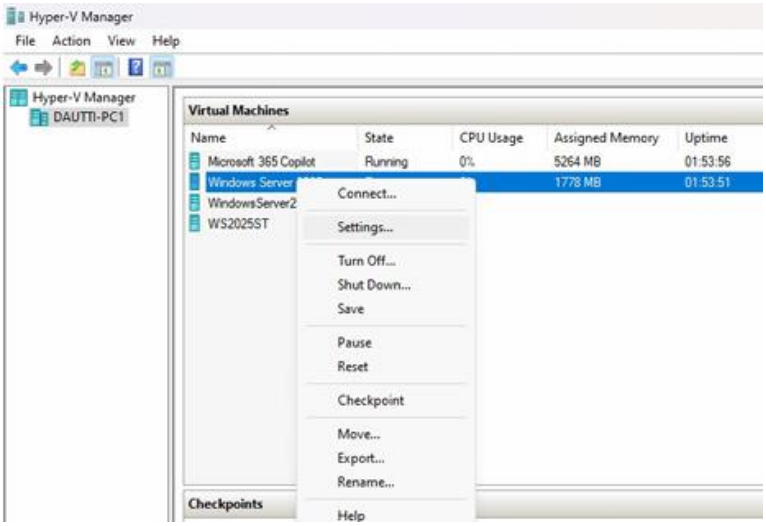

Рисунок 11.12 – Налаштування віртуальної машини [13]

У вікні налаштувань у розділі «Апаратне забезпечення» обирається «Пам’ять», де доступні відповідні інструменти керування (рис. 11.13).

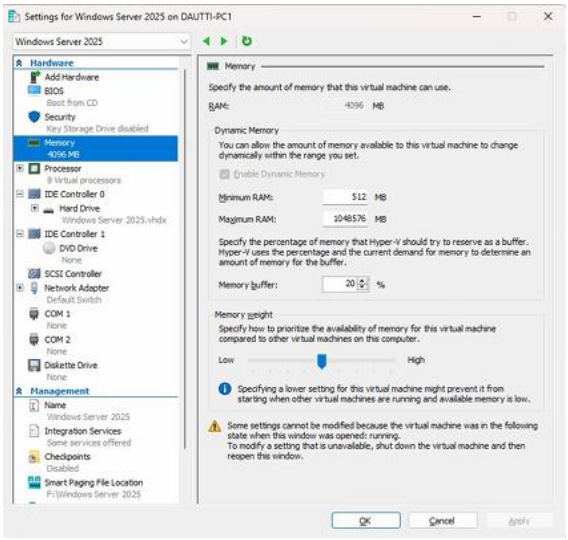

Рисунок 11.13 – Керування віртуальною пам’яттю [13]

Існує дві опції конфігурації: виділення фіксованого обсягу пам’яті шляхом введення бажаного розміру в мегабайтах, або використання динамічної пам’яті шляхом встановлення прапорця «Увімкнути динамічну пам’ять». Остання функція дозволяє Hyper-V розподіляти пам’ять на основі попиту ВМ, встановлюючи мінімальне та максимальне значення RAM. Варто також згадати концепцію надлишкового виділення ресурсів, яка стосується практики виділення більшої кількості віртуальних ресурсів, ніж може підтримати фізичне обладнання. Хоча це може підвищити гнучкість, це також несе ризики зниження продуктивності, тому балансування розподілу ресурсів є критично важливим [15].

Налаштування віртуальної мережі є необхідним для забезпечення зв’язку між ВМ та із зовнішньою мережею. У Hyper-V це досягається шляхом конфігурації віртуального комутатора. Розрізняють три основні типи: зовнішній комутатор (підключає ВМ до фізичного мережевого адаптера хоста, дозволяючи доступ до зовнішньої мережі), внутрішній комутатор (підключає ВМ до хоста, але не надає доступу до зовнішньої мережі) та приватний комутатор (дозволяє зв’язок виключно між ВМ на одному хості). Для налаштування використовується «Диспетчер віртуальних комутаторів...» (Virtual Switch Manager...) на панелі дій (рис. 11.14).

Рисунок 11.14 – Створення віртуального комутатора (Virtual Switch Manager) [13]

Обирається тип комутатора, натискається «Створити віртуальний комутатор», після чого налаштовуються його параметри (рис. 11.15).

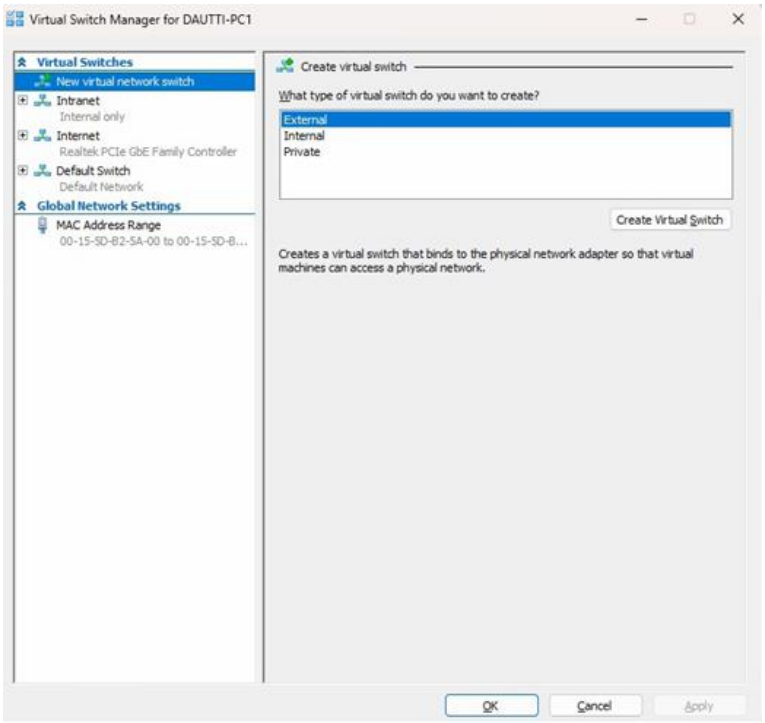

Рисунок 11.15 – Властивості віртуального комутатора [13]

Вказується ім’я та тип підключення. За необхідності вмикається ідентифікація віртуальної локальної мережі (VLAN) для керування трафіком та підвищення безпеки.

Контрольні точки в Hyper-V є ключовою функцією, що дозволяє захопити та зберегти стан ВМ у певний момент часу. Це є неоціненним для підтримки стабільності системи під час критичних операцій. Для створення контрольної точки необхідно натиснути правою кнопкою миші на ВМ і вибрати «Контрольна точка» (Checkpoint) (рис. 11.16).

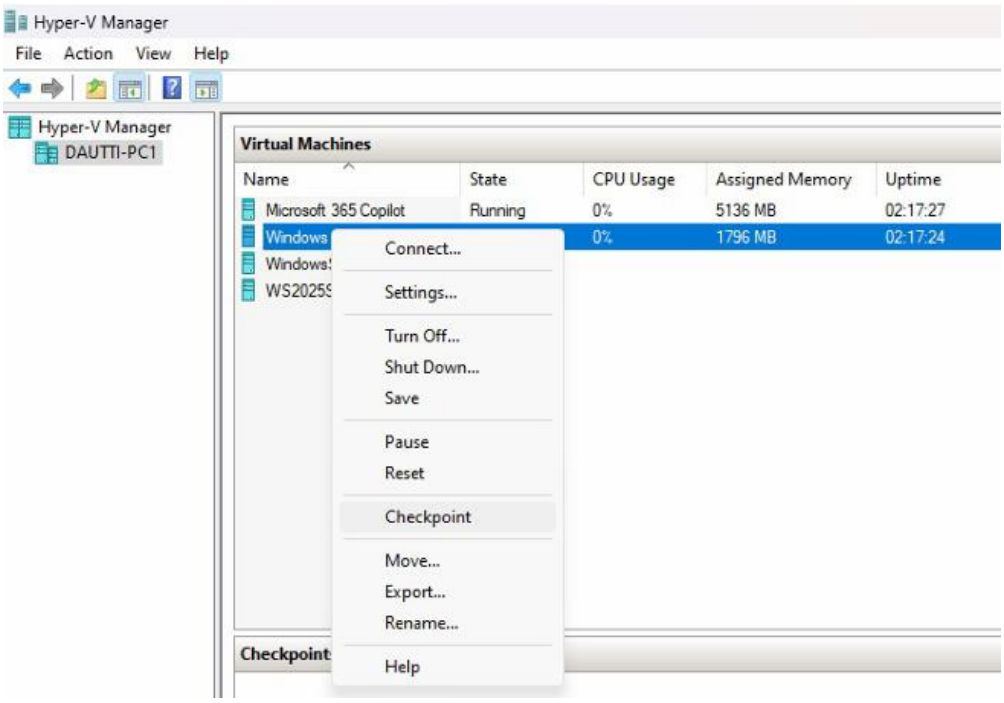

Рисунок 11.16 – Створення контрольної точки (Checkpoint) [13]

Після ініціювання процесу система може відобразити підтвердження успішного виконання операції (рис. 11.17).

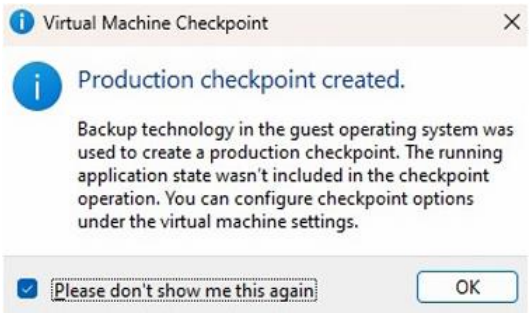

Рисунок 11.17 – Підтвердження створення контрольної точки [13]

Hyper-V пропонує два типи контрольних точок: виробнича контрольна точка (Production Checkpoint), яка фокусується на захопленні стану ВМ з точки зору операційної системи без включення стану запущених додатків, та стандартна контрольна точка (Standard Checkpoint), яка захоплює повний стан ВМ, включаючи всі запущені додатки (рис. 11.18).

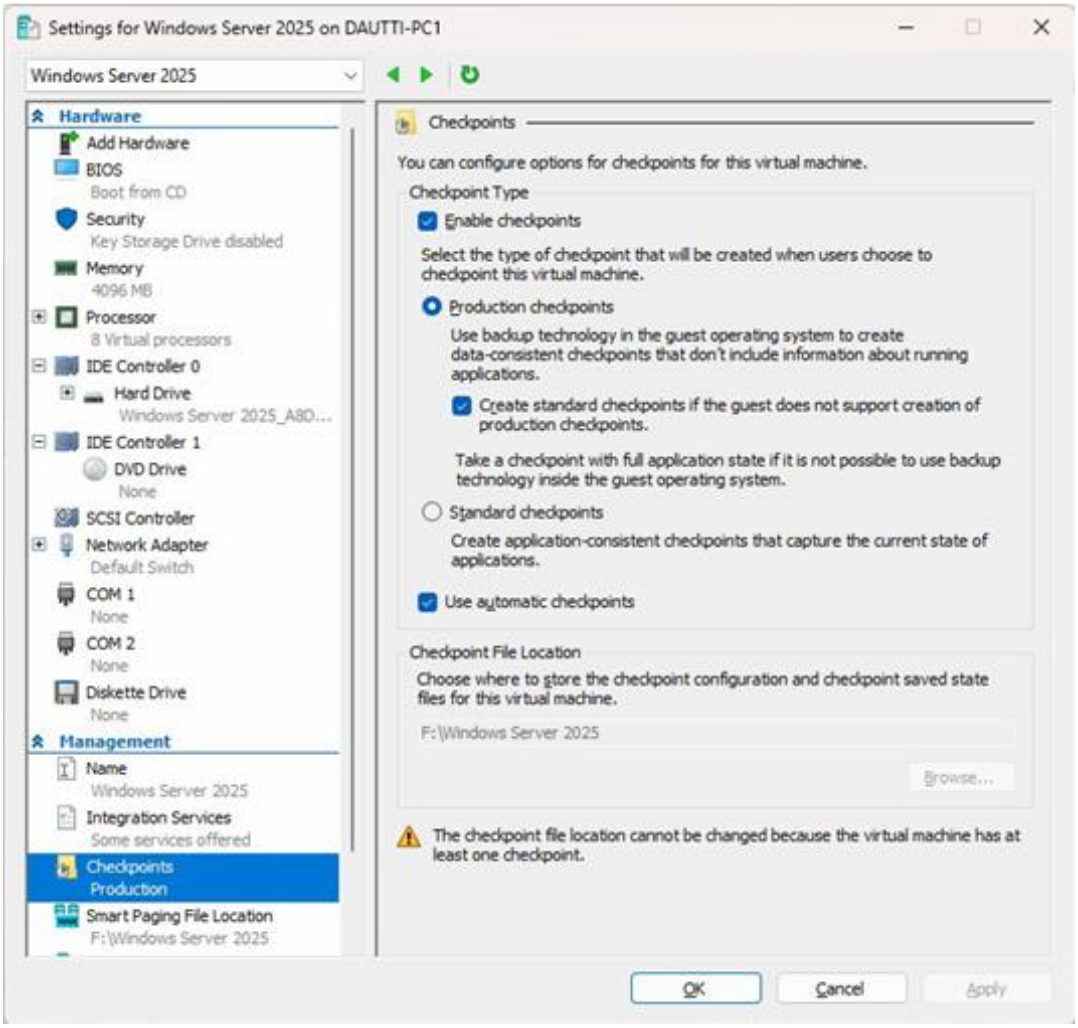

Рисунок 11.18 – Типи контрольних точок (Production та Standard) [13]

З моменту дебюту у Windows Server 2008 можливості віртуального дискового сховища Hyper-V значно еволюціонували. Спочатку використовувався формат VHD з обмеженням розміру до 2 ТБ. З введенням Windows Server 2012 було представлено формат VHDX, який збільшив ліміт до 64 ТБ, покращив стійкість до збоїв живлення та продуктивність. Незважаючи на ці досягнення, формат VHD залишається підтримуваним у Windows Server 2025 для сумісності [13].

Конвертація фізичних серверів у віртуальні машини (P2V) реалізується за допомогою інструменту Disk2vhd від Microsoft, який перетворює фізичні дискові приводи у файли VHD (рис. 11.19).

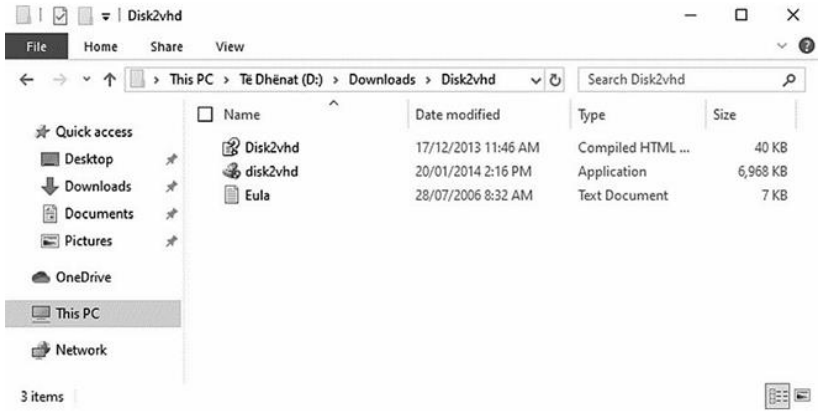

Рисунок 11.19 – Додаток Disk2vhd для конвертації фізичного диска у VHD [13]

Після генерації VHD можна використовувати Диспетчер Hyper-V для налаштування нової ВМ. Зворотний процес, відомий як конвертація з віртуального у фізичне середовище (V2P), є менш підтримуваним і часто вимагає сторонніх рішень або ручної міграції. Для виконання V2P конвертації можна використовувати програмне забезпечення для клонування, таке як EZ Gig IV.

Перехід від VMware до Hyper-V вимагає ретельного планування. Ключові міркування включають оцінку поточного середовища, перевірку сумісності та ліцензування, а також резервне копіювання даних. Microsoft надає інструменти для полегшення міграції, такі як Microsoft Virtual Machine Converter (MVMC) та Disk2VHD. Процес міграції включає підготовку середовища Hyper-V, конвертацію ВМ, тестування та фіналізацію налаштувань мережі [13].

Для ефективного керування ВМ у Hyper-V доступ до її налаштувань здійснюється через контекстне меню «Параметри» (рис. 11.20).

Вікно конфігурації дозволяє налаштувати такі аспекти: додавання обладнання (Add Hardware), прошивка (Firmware), BIOS (порядок завантаження), безпека (Security) для шифрування, пам’ять (Memory), процесор (Processor), контролери IDE та SCSI для керування накопичувачами, мережевий адаптер (Network Adapter), порти COM та дисковод для гнучких дисків. Ретельне налаштування цих параметрів дозволяє оптимізувати продуктивність ВМ та адаптувати її до специфічних операційних вимог.

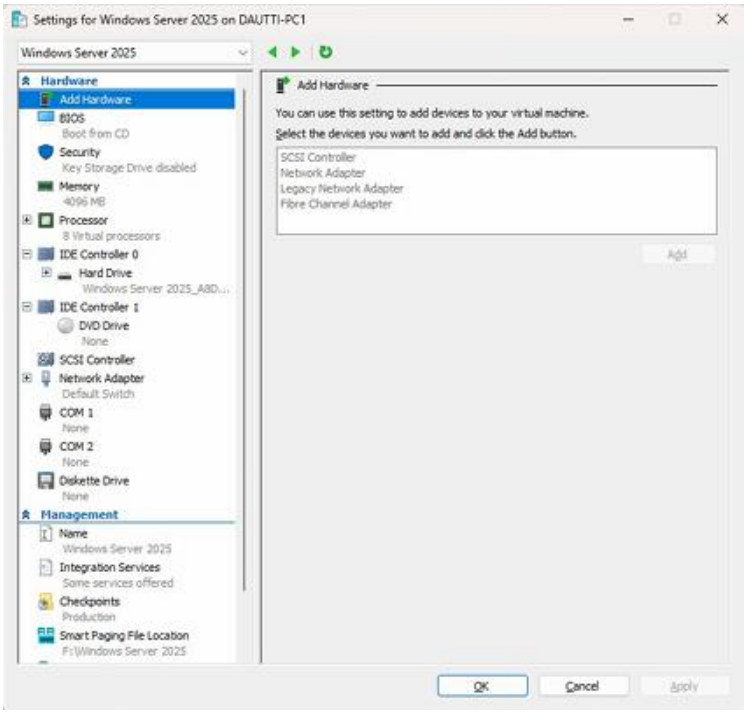

Рисунок 11.20 – Встановлення параметрів ВМ [13]

Під час керування віртуальними машинами у Hyper-V використання як панелі дій, так і контекстного меню ВМ дозволяє суттєво оптимізувати адміністративні завдання. Панель дій, зображена на рисунку 11.21, є важливим компонентом Диспетчера Hyper-V, що полегшує комплексне керування ВМ. Вона надає опції для створення нових віртуальних машин, конфігурації параметрів Hyper-V, налаштування віртуальних комутаторів та створення віртуальних мереж зберігання даних (SAN). Ця панель також дозволяє здійснювати модифікацію та перевірку віртуальних дисків, зупинку та запуск служб, видалення ВМ та оновлення списку доступних ВМ. Її роль як центрального інструменту керування гарантує, що адміністратори можуть виконувати широкий спектр завдань з єдиного інтерфейсу, підвищуючи ефективність та зручність використання.

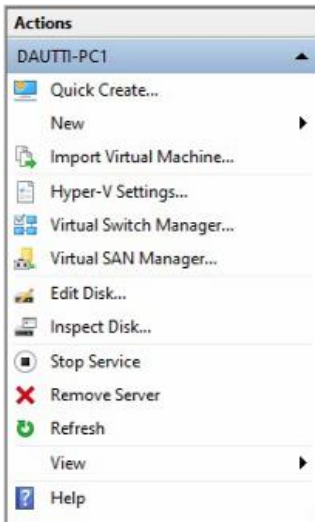

Рисунок 11.21 – Панель дій у Диспетчері Hyper-V [13]

Контекстне меню ВМ, проілюстроване на рисунку 11.22, доповнює панель дій, пропонуючи опції, специфічні для вибраної ВМ. Це меню включає такі важливі функції, як «Підключити...» (Connect...) для доступу до консолі ВМ, «Перейменувати...» (Rename...) для оновлення імені ВМ, а також різні інші опції керування, адаптовані до окремої ВМ. Наприклад, можна керувати налаштуваннями ВМ, контрольними точками та знімками або навіть контролювати стан її живлення (наприклад, запуск, зупинка, пазу) безпосередньо з цього меню. Специфічність контекстного меню дозволяє здійснювати точне, сфокусоване керування окремими ВМ, що робить його цінним інструментом для виконання завдань, специфічних для ВМ. Розуміння того, як використовувати як панель дій, так і контекстне меню, озброює адміністратора надійним набором інструментів для ефективного керування ВМ у Hyper-V.

Рисунок 11.22 – Контекстне меню в Диспетчері Hyper-V [13]

Налаштування параметрів запуску та відновлення ВМ є важливим для забезпечення безперебійної роботи та стійкості системи, особливо після перезавантаження хоста. Ці налаштування дозволяють адміністраторам контролювати поведінку ВМ у відповідь на перезапуск хоста, мінімізуючи потенційний час простою та оптимізуючи розподіл ресурсів. Рекомендується конфігурувати «Дію при запуску» (Startup Action) на основі операційних потреб. Опція «Нічого не робити» (Do Nothing) ідеально підходить для некритичних ВМ, що може допомогти зберегти ресурси хоста після перезапуску. Опція «Автоматично запускати, якщо вона працювала» (Automatically Start if Running) призначена для важливих ВМ, забезпечуючи відновлення їхнього попереднього стану без необхідності ручного втручання, що є корисним для послідовної безперервності надання послуг. Опція «Завжди запускати» (Always Start) підходить для критичних систем, які повинні бути оперативними негайно після перезапуску хоста, незалежно від їхнього попереднього стану [13].

Важливим параметром є також «Затримка автоматичного запуску» (Automatic Start Delay). Поетапний запуск ВМ може запобігти вузьким місцям у продуктивності шляхом зменшення миттєвого навантаження на центральний процесор та пам’ять. Ця функція є корисною в середовищах з декількома ВМ на одному хості. Окрім того, налаштування «Дії при автоматичній зупинці» (Automatic Stop Action) для коректного завершення роботи ВМ при вимкненні або перезавантаженні хоста допомагає запобігти втраті даних та підтримує цілісність ВМ. Hyper-V надає опції в налаштуваннях автоматичної дії при запуску та зупинці для кожної віртуальної машини, дозволяючи адміністраторам конфігурувати поведінку ВМ при запуску або вимкненні хоста Hyper-V. Ці налаштування є важливими в середовищах Hyper-V для забезпечення часу безвідмовної роботи для критичних робочих навантажень шляхом автоматичного перезапуску важливих ВМ, керування розподілом ресурсів під час перезавантаження хоста шляхом встановлення затримок для некритичних ВМ, що допомагає запобігти вузьким місцям у продуктивності, а також збереження цілісності даних ВМ шляхом налаштування дії завершення роботи для вимкнення ВМ перед коректним вимкненням хоста. Ці конфігурації запуску та відновлення є невід’ємною частиною опцій керування Hyper-V та відіграють ключову роль у підтримці надійних, стійких операцій ВМ. Дотримання цих практик забезпечує добре організоване середовище ВМ з мінімізованим впливом від несподіваних перезавантажень та оптимізованою продуктивністю робочих навантажень.

Для покращення розуміння та надання практичних висновків у цьому розділі висвітлюються реальні сценарії, де Hyper-V відіграє вирішальну роль у сучасних IT-операціях. Ці приклади не лише демонструють універсальність Hyper-V, але й пропонують дієві кроки для загальних галузевих практик, таких як міграція, аварійне відновлення, автоматизація та резервне копіювання.

Для багатьох організацій міграція з VMware на Hyper-V представляє стратегічний зсув у напрямку консолідації IT-ресурсів в екосистемі Microsoft. Ця міграція вимагає продуманого планування, починаючи з комплексної оцінки сумісності існуючих ВМ. Використання таких інструментів, як Microsoft Virtual Machine Converter (MVMC) або System Center Virtual Machine Manager (SCVMM), може оптимізувати процес міграції шляхом автоматизації певних етапів, таких як конвертація дисків ВМ та конфігурація мережі. Тестування кожної ВМ у проміжному середовищі перед реальним розгортанням забезпечує оптимальну функціональність та зменшує ризик потенційних проблем. Завдяки ретельній підготовці організації можуть перейти на Hyper-V, підтримуючи високу продуктивність та мінімізуючи перебої в обслуговуванні.

Hyper-V Replica є потужною функцією для аварійного відновлення, що дозволяє організаціям реплікувати ВМ на вторинний майданчик, як локально, так і в хмарі. Це налаштування забезпечує критичну мережу безпеки, гарантуючи швидке відновлення та мінімальну втрату даних у разі збою основного майданчика. Конфігурація Hyper-V Replica передбачає налаштування реплікації на рівні ВМ, визначення частоти реплікації на основі цільових точок відновлення (Recovery Point Objectives – RPO) та конфігурацію мережевих з’єднань між основним та реплікаційним майданчиками. Регулярно реплікуючи ВМ на резервний майданчик, бізнес може захистити свої дані та зменшити час простою, підтримуючи стійку інфраструктуру [13].

Рутинне технічне обслуговування, таке як оновлення системи, може вносити зміни, що впливають на стабільність ВМ. Автоматизація контрольних точок ВМ за допомогою PowerShell перед кожним оновленням є найкращою практикою для полегшення відкату у разі виникнення проблем. Наступний скрипт PowerShell створює контрольну точку для кожної запущеної ВМ, позначаючи кожен знімок міткою часу для легкої ідентифікації.

Цей скрипт допомагає адміністраторам економити час, впроваджуючи механізм безпеки для декількох ВМ, сприяючи операційній узгодженості та мінімізуючи ризик, пов'язаний з оновленнями.

Регулярне резервне копіювання ВМ Hyper-V є необхідним для безперервності бізнесу, дотримання нормативних вимог та захисту даних. Windows Server 2025 надає такі інструменти, як Windows Server Backup та System Center Data Protection Manager (DPM), для планування автоматизованого резервного копіювання або створення знімків за вимогою. Налаштування рутинного резервного копіювання гарантує, що стан ВМ, конфігурація та дані надійно зберігаються, підтримуючи швидке відновлення у разі випадкової втрати даних або кіберінцидентів. Впроваджуючи регулярне резервне копіювання, організації не лише захищають критичні дані, але й будують стійку та відповідну вимогам IT-інфраструктуру, здатну задовольнити сучасні вимоги бізнесу. Ці реальні приклади підкреслюють можливості Hyper-V у досягненні надійних та ефективних віртуалізованих середовищ, допомагаючи IT-фахівцям застосовувати ці найкращі практики безпосередньо у своїх організаціях. З цим фундаментом здійснюється підготовка до переходу до практичних вправ, таких як інсталяція ролі Hyper-V у Windows Server 2025, для подальшого вдосконалення навичок та застосування знань у реальних сценаріях [15].

Технічні вимоги

Впровадження платформи віртуалізації Hyper-V вимагає дотримання чітко визначених специфікацій обладнання, причому для окремих функціональних можливостей можуть висуватися додаткові умови. Наведена нижче інформація призначена для аналізу відповідності системи встановленим критеріям, що є необхідним для планованої експлуатації середовища.

Незалежно від обраного набору функцій Hyper-V, базові вимоги до апаратної платформи є обов'язковими для виконання. Ключовою умовою є наявність 64-розрядного процесора з підтримкою технології трансляції адрес другого рівня (SLAT). Наявність SLAT є критичною для інсталяції компонентів віртуалізації, таких як гіпервізор Windows. Водночас, для встановлення засобів керування Hyper-V, до яких належать «Підключення до віртуальної машини» (Virtual Machine Connection або VMConnect), Диспетчер Hyper-V та командлети Hyper-V для Windows PowerShell, наявність SLAT на процесорі не вимагається. Окрім цього, необхідною є підтримка розширень режиму моніторингу віртуальної машини. Окремої уваги потребує підсистема оперативної пам'яті. Планування ресурсів має виходити з мінімального обсягу у 4 ГБ, при цьому більший обсяг пам'яті забезпечує кращу продуктивність. Системі має бути доступний достатній обсяг пам'яті як для функціонування хоста, так і для одночасної роботи всіх запланованих віртуальних машин [16].

На рівні базової системи вводу-виводу (BIOS) або інтерфейсу UEFI повинна бути активована підтримка віртуалізації. Це передбачає наявність апаратної віртуалізації, що реалізується у процесорах з відповідними опціями, зокрема Intel Virtualization Technology (Intel VT) або AMD Virtualization (AMD-V).

Також обов'язковою вимогою є доступність та активація апаратного запобігання виконанню даних (DEP). Для архітектури Intel ця технологія позначається як біт XD (Execute Disable Bit), а для систем AMD – як біт NX (No Execute Bit) [16].

Для верифікації відповідності апаратного забезпечення вимогам Hyper-V застосовуються інструменти командного рядка. Процедура передбачає відкриття Windows PowerShell або командного рядка та введення команди Systeminfo.exe (рис. 11.23).

У згенерованому звіті необхідно прокрутити вміст до розділу вимог Hyper-V. Якщо всі перелічені параметри мають значення «Так», система придатна для запуску ролі Hyper-V. У разі, якщо будь-який з елементів повертає значення «Ні», вимагається перегляд вимог, наведених у документації, та внесення відповідних апаратних або конфігураційних коректив.

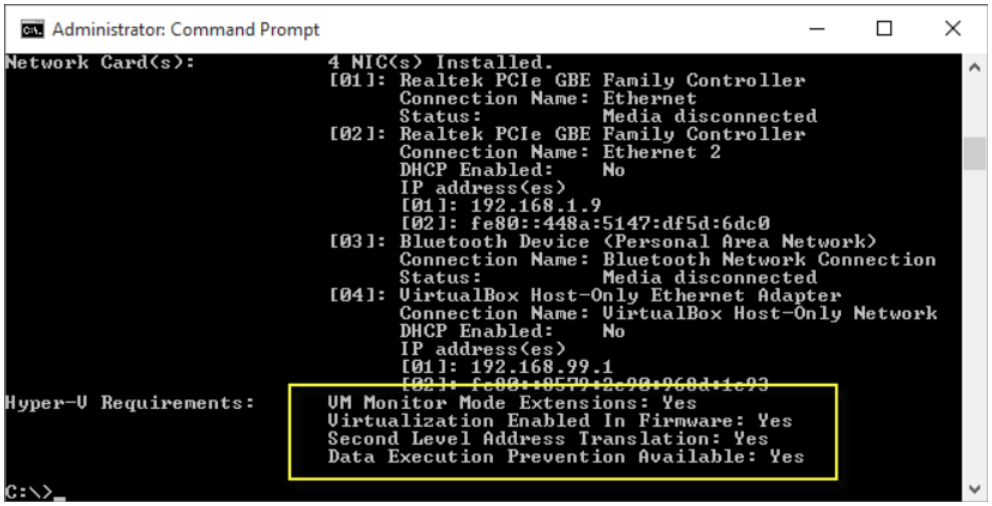

Рисунок 11.23 – Звіт команди Systeminfo.exe [16]

Окремий набір вимог висувається до таких функцій, як призначення дискретних пристроїв та використання екранованих віртуальних машин. Стосовно призначення дискретних пристроїв (Discrete Device Assignment), вимоги до хоста аналогічні тим, що встановлені для функції SR-IOV у Hyper-V. Процесор повинен підтримувати роботу з розширеною таблицею сторінок (EPT) для архітектури Intel або вкладеною таблицею сторінок (NPT) для архітектури AMD. Системна логіка (чіпсет) повинна забезпечувати перепризначення переривань, що реалізується через Intel VT-d з можливістю перепризначення переривань (VT-d2) або будь-яку версію блоку керування пам'яттю вводу/виводу AMD (I/O MMU). Також необхідна підтримка перепризначення прямого доступу до пам'яті (DMA), що забезпечується Intel VT-d з черговими інвалідаціями або I/O MMU AMD. Обов'язковою є наявність служб контролю доступу (ACS) на кореневих портах PCI Express [16].

Таблиці прошивки системи повинні бути сконфігуровані таким чином, щоб надавати гіпервізору Windows доступ до I/O MMU. Зазначена функція може бути деактивована в налаштуваннях UEFI або BIOS, тому для її активації рекомендується звернутися до документації обладнання або виробника. Стосовно пристроїв, що призначаються, вимагається наявність графічного процесора або енергонезалежної пам'яті Express (NVMe). Слід зауважити, що лише певні моделі графічних пристроїв підтримують призначення дискретних пристроїв, що потребує перевірки через технічну документацію. Деталізовану інформацію про використання цієї функції та важливі аспекти налаштування можна знайти у спеціалізованих джерелах, присвячених опису та довідці щодо призначення дискретних пристроїв.

У випадку успішного виконання всіх вимог до операційної системи, апаратного забезпечення та сумісності, у відповідному інтерфейсі керування стає доступним розділ Hyper-V.

Розуміння розподілу дисків і параметрів зберігання даних

Встановлення нових операційних систем класифікується як рутинне завдання в межах системного адміністрування. Цей процес охоплює низку критичних етапів, зокрема підготовку інсталяційного носія, безпосереднє виконання інсталяції ОС, перевірку результатів розгортання та налаштування початкової конфігурації сервера. Зазначені кроки є фундаментальними для формування бази подальших операцій. Хоча певні сервери можуть постачатися з попередньо встановленими операційними системами, експертиза системного адміністратора часто є необхідною для забезпечення відповідності встановленої ОС специфічним потребам інфраструктури. Перед початком процесу інсталяції розглядається важливість схем розподілу для організації дискових розділів [16].

Розподіл диска визначається як процес поділу фізичного носія на логічні секції, відомі як розділи. Кожен розділ може функціонувати під управлінням окремої файлової системи, наприклад, файлової системи нової технології (NTFS) або стійкої файлової системи (ReFS), і використовуватися для зберігання різноманітних типів даних. Розділи також можуть

застосовуватися для створення окремих томів – логічних одиниць зберігання, що можуть охоплювати кілька фізичних дисків [13].

Схема розподілу є технікою, що детермінує спосіб створення та керування цими розділами на дисках. Виділяють дві основні схеми розподілу: Головний завантажувальний запис (MBR) та Таблицю розділів GUID (GPT). MBR – це старіша схема розподілу, яка наразі вважається застарілою і не рекомендується для сучасних систем. MBR оперує дисковими секторами розміром 512 байт і підтримує лише 4 основні розділи або 1 розширений розділ, що містить до 26 логічних розділів. Для керування дисками використовується логічна адресація блоків (LBA) з максимальним обмеженням обсягу в 2 ТБ. Хоча MBR свого часу була корисною для систем із кількома варіантами завантаження, вона має низку обмежень, що робить її несумісною із сучасними технологіями: обмеження розміру в 2 ТБ є недостатнім для багатьох сучасних пристроїв, а лімітована кількість розділів може стати вузьким місцем для складних конфігурацій [13].

Додатково MBR позбавлена розширених функцій надлишковості та відновлення, властивих новішим схемам. Ці недоліки зумовили розробку Таблиці розділів GUID (GPT) – сучасної схеми, що долає недоліки MBR. GPT використовує 128-бітний глобальний унікальний ідентифікатор (GUID) для ідентифікації ресурсів. Підтримуються розміри блоків від 512 байт і вище, із загальноприйнятим стандартом у 4096 байт, де кожен запис розділу займає 128 байт. GPT є частиною стандарту єдиного розширюваного інтерфейсу прошивки (UEFI), який замінює застарілий BIOS. Ця схема характеризується стійкістю, здатністю обробляти до 9,4 зетабайт (ZB) дискового простору та підтримувати до 128 розділів на диск. Також GPT забезпечує кращі функції надійності та безпеки, такі як захисний MBR та контрольна сума CRC32. Важливо розуміти сутність CRC32 – алгоритму контрольної суми, що використовується для виявлення помилок при передачі або зберіганні даних. CRC32 генерує унікальне значення, похідне від вмісту даних, яке порівнюється з оригінальною сумою для перевірки цілісності. Невідповідність значень свідчить про можливе пошкодження даних під час передачі чи зберігання.

Для інсталяції Windows Server 2025 використання схеми розділів GPT є обов’язковим. Ця необхідність випливає з вимоги використання UEFI, який замінює традиційний BIOS і здійснює завантаження виключно з дисків GPT. UEFI не лише забезпечує швидші та безпечніші процеси завантаження, але й підтримує розширені функціональні можливості, такі як безпечне завантаження (Secure Boot) та BitLocker. Secure Boot захищає процес завантаження, дозволяючи запуск лише авторизованого ПЗ та блокуючи шкідливий код, наприклад руткіти. BitLocker доповнює це повним шифруванням диска, гарантуючи безпеку даних навіть у разі фізичного вилучення пристрою [17].

У Windows Server 2025 створення та керування розділами здійснюється через інструмент «Керування дисками» або утиліту Diskpart. Альтернативно використовується майстер налаштування Windows під час інсталяції. При необхідності конвертації MBR у GPT попередньо видаляються всі розділи, тому критично важливо виконати резервне копіювання даних. Окрім вибору схеми розділів, налаштовуються параметри завантаження (Boot settings) в BIOS або UEFI для визначення джерела запуску ОС (DVD, USB, мережа). Для створення завантажувального USB-накопичувача можна використовувати інструмент Windows 7 USB/DVD Download Tool, доступний на офіційному сайті Microsoft [17].

Окрім розподілу дисків, Windows Server 2025 пропонує різноманітні варіанти зберігання для підвищення продуктивності, доступності та масштабованості. До ключових функцій належать Storage Spaces, Storage Spaces Direct та Storage Replica. Функція Storage Spaces дозволяє створювати віртуальні диски з пулу фізичних дисків, пропонуючи різні рівні стійкості (простий, дзеркальний, з парністю) та підтримує багаторівневе зберігання для автоматичного переміщення даних між SSD та HDD.

Storage Spaces Direct уможливлює формування спільного пулу зберігання з локальних дисків у межах кластера, сприяючи створенню рішень гіперконвергентної інфраструктури (HCI) або програмно-визначеного сховища (SDS). HCI представляє структуру, що об’єднує обчислення, зберігання та мережу в єдину систему, керовану програмним забезпеченням, що оптимізує керування та масштабованість. SDS, у свою чергу, відокремлює обладнання для

зберігання від програмного забезпечення керування, дозволяючи динамічно розподіляти ресурси.

Функція Storage Replica забезпечує реплікацію даних між серверами або кластерами (синхронну або асинхронну), уможливлюючи створення розтягнутих кластерів або реплікацію між сайтами. Розтягнуті кластери розподіляють єдиний кластер між кількома локаціями для забезпечення безперебійної роботи, тоді як реплікація між сайтами синхронізує дані між [13].

Після завершення інсталяції важливим є розуміння розширених параметрів запуску. У Windows Server 2025 відсутня опція використання клавіші F8 для відновлення ОС. Натомість доступ до розширених опцій здійснюється через меню налаштувань. Процес ініціюється натисканням кнопки «Пуск», вибором піктограми «Налаштування» (Settings), переходом до розділу «Система» (System) та вибором опції «Відновлення» (Recovery). У правій частині екрана, у розділі параметрів відновлення, натискається кнопка «Перезавантажити зараз» (Restart now), як показано на рисунку 11.24.

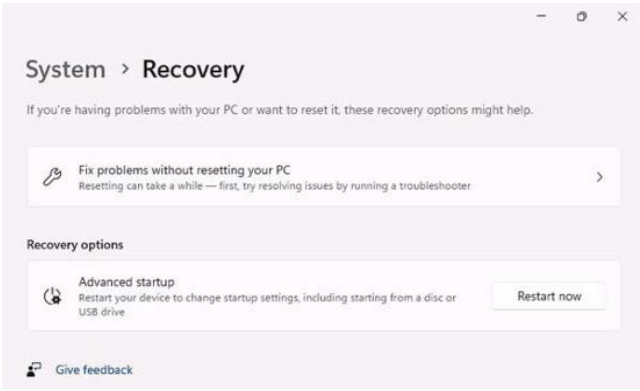

Рисунок 11.24 – Перехід до розширеного запуску в Windows Server 2025 [13]

Після цього з’являється діалогове вікно попередження про збереження роботи, де необхідно повторно натиснути «Перезавантажити зараз» та обрати причину дії. Після перезавантаження системи на екрані вибору опцій обирається пункт «Виправлення неполадок» (Troubleshoot). На екрані розширених опцій, як показано на рисунку 11.25, стає можливим вибір різноманітних інструментів для відновлення або ремонту серверної операційної системи.

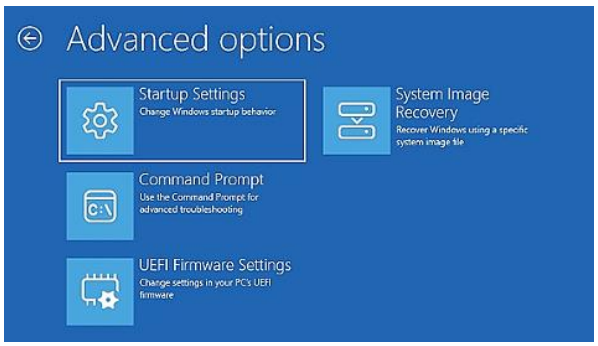

Рисунок 11.25 – Розширені параметри (Advanced Options) у Windows Server 2025 [13]

Квоти

У контексті адміністрування серверів під керуванням Windows Server 2025 критично важливим завданням є контроль використання дискового простору користувачами та додатками. Для вирішення цієї задачі застосовується механізм дискових квот, який дозволяє обмежувати обсяг даних, що зберігаються, та запобігати переповненню фізичних носіїв, що могло б призвести до відмови в обслуговуванні критичних сервісів.

У сучасній версії Windows Server 2025 розрізняють два основні підходи до реалізації квотування: стандартні квоти NTFS та розширені квоти диспетчера ресурсів файлового сервера (File Server Resource Manager – FSRM). Стандартні квоти NTFS функціонують на

рівні логічного тому і прив'язуються до облікового запису користувача, який є власником файлів. Цей метод є базовим і дозволяє встановити ліміт дискового простору для конкретного користувача на всьому диску, незалежно від розташування файлів у папках. Хоча цей підхід забезпечує загальний контроль, він характеризується недостатньою гнучкістю для складних корпоративних сценаріїв, де необхідно обмежувати розмір конкретних спільних папок, а не загальний простір користувача [13].

Для більш гранулярного та ефективного керування в Windows Server 2025 рекомендується використання Диспетчера ресурсів файлового сервера (FSRM). На відміну від квот NTFS, квоти FSRM застосовуються безпосередньо до папок або томів, що дозволяє адміністраторам контролювати розмір директорій спільного доступу, незалежно від того, хто є власником файлів у них.

Система FSRM підтримує два типи квот: жорсткі та м'які. Жорстка квота фізично забороняє запис даних на диск після досягнення встановленого ліміту, генеруючи помилку про недостатність місця для користувача. Цей тип використовується для суворого дотримання політик зберігання. М'яка квота, навпаки, не блокує запис даних при перевищенні ліміту, а використовується виключно для моніторингу та сповіщення. Вона дозволяє адміністраторам відстежувати тенденції зростання даних та отримувати повідомлення про порушення політик використання простору, не перериваючи робочий процес користувачів [13].

Процес налаштування квот у FSRM оптимізується за допомогою використання шаблонів квот. Шаблони визначають набір стандартних параметрів, таких як ліміт простору, тип квоти (жорстка чи м'яка) та порогові значення сповіщень.

Використання шаблонів дозволяє централізовано керувати політиками: при зміні параметрів у шаблоні зміни можуть автоматично поширюватися на всі квоти, що базуються на ньому. Важливим елементом конфігурації є налаштування порогових значень, які ініціюють дії при заповненні квоти на певний відсоток (наприклад, 85%, 95% або 100%). Дії можуть включати надсилання електронних листів адміністратору або користувачу, запис події в журнал Windows, виконання скриптів або формування звітів. Такий підхід забезпечує проактивне керування інфраструктурою зберігання даних, дозволяючи виявляти та вирішувати проблеми з нестачею вільного місця до настання критичних ситуацій.

Дослідження конфігурацій завантаження та параметрів запуску

Перед завантаженням операційної системи комп'ютер повинен пройти процес ініціалізації, який охоплює перевірку апаратних компонентів та завантаження системного програмного забезпечення. Цей процес керується мікропрограмним забезпеченням – BIOS або UEFI, залежно від архітектури материнської плати та апаратного забезпечення. Як BIOS, так і UEFI відповідають за конфігурацію параметрів завантаження, включаючи порядок завантаження, режим завантаження та вибір пріоритетного пристрою. Параметри завантаження мають суттєвий вплив на продуктивність сервера та його взаємодію з іншими пристроями й мережами. Розуміння відмінностей між BIOS та UEFI, а також їхніх відповідних переваг і недоліків, є критично важливим для системного адміністрування. У цьому розділі розглядаються опції завантаження, доступні в BIOS та UEFI, а також методологія їх налаштування в середовищі Windows Server 2025.

Для коректного запуску системи необхідно розуміти опції завантаження в UEFI (Unified Extensible Firmware Interface), який значною мірою витіснив застарілий BIOS у сучасних системах. UEFI визначається як інтерфейс мікропрограми, що ініціалізує обладнання та ефективно завантажує операційну систему. Доступ до налаштувань UEFI здійснюється під час запуску шляхом натискання певних клавіш (F2, F10, Delete або Esc), які варіюються залежно від виробника. На відміну від старіших систем, UEFI пропонує розширені функції, такі як безпечне завантаження, прискорений час завантаження та підтримку жорстких дисків великого обсягу з використанням GPT.

Для забезпечення безперебійного процесу інсталяції Windows Server 2025 критично важливо налаштувати порядок завантаження, встановивши інсталяційний носій (USB або DVD) як основний пристрій завантаження. Додатково настійно рекомендується активація безпечного режиму для підвищення рівня безпеки. Ця функція дозволяє завантажувати лише

довірене програмне забезпечення, запобігаючи змінам з боку шкідливого ПЗ. Оскільки Secure Boot часто вимагається стандартами безпеки, необхідно використовувати диск із розміткою GPT, адже ця функція не підтримує розділи MBR. Попередня перевірка цих конфігурацій дозволяє мінімізувати ймовірність збоїв інсталяції через невідповідність налаштувань [20].

При вході в інтерфейс BIOS стають доступними різні варіанти завантаження. Одним із поширених типів інсталяційного носія є завантажувальний DVD. Для його використання необхідно налаштувати комп'ютер на завантаження з DVD-приводу, змінивши порядок завантаження в налаштуваннях BIOS та встановивши оптичний привід як пріоритетний пристрій. Іншим методом є використання завантажувального USB-накопичувача, який повинен мати обсяг не менше 8 ГБ. Процедура вимагає підключення накопичувача та вибору його як першої опції у послідовності завантаження в BIOS. Також існує метод мережевого завантаження (PXE Boot), що дозволяє завантажувати інсталяційні файли з віддаленого сервера через локальну мережу (LAN). Для цього в налаштуваннях BIOS активується опція мережевого завантаження та встановлюється відповідний пріоритет. Вибір методу залежить від уподобань адміністратора та наявності ресурсів [15].

Розуміння роботи апаратних компонентів та процесу запуску дозволяє технічним спеціалістам швидко вирішувати проблеми та скорочувати час простою сервера. Коли сервер вмикається, початкова активність пов'язана з чіпом на материнській платі, відомим як ПЗУ (ROM), який активує програму BIOS. BIOS відіграє ключову роль у керуванні функціональністю обладнання, виявляючи та конфігуруючи такі компоненти, як центральний процесор (CPU), пам'ять та диски. Крім того, BIOS ідентифікує завантажувальні пристрої, визначаючи джерела ініціації процесу завантаження. Цей процес проілюстровано на рисунку 11.26.

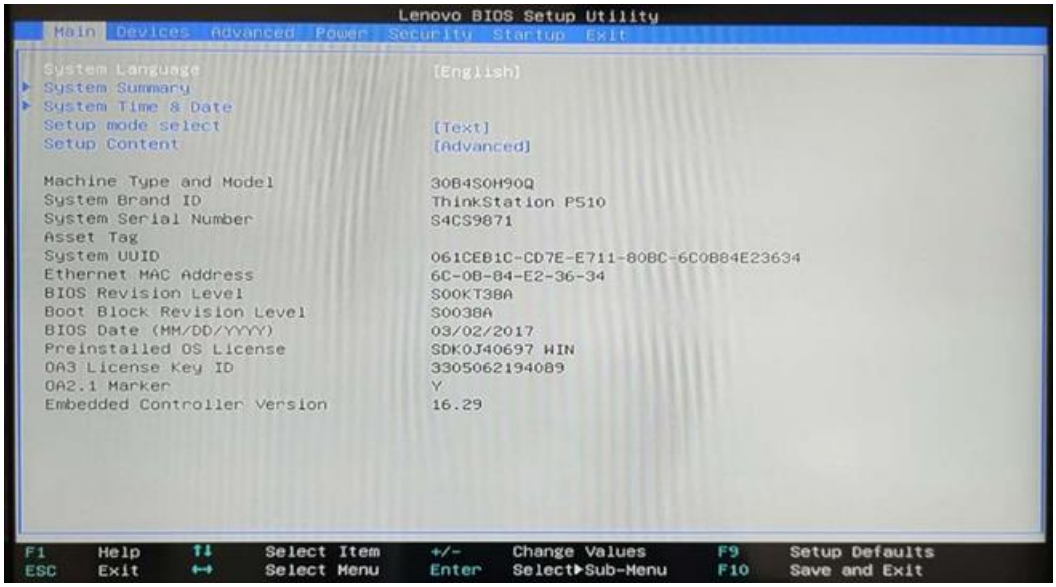

Рисунок 11.26 – Екран конфігурації BIOS [13]

Однак BIOS має обмеження, що робить його недостатнім для сучасних серверів. Для вирішення цих проблем було розроблено UEFI, який забезпечує швидше та безпечніше завантаження, підтримку більших дисків та покращений графічний інтерфейс.

Сучасні обчислювальні системи використовують UEFI, розроблений консорціумом UEFI для подолання обмежень BIOS. UEFI може працювати в 32-розрядних та 64-розрядних режимах процесора, маючи доступ до всього обсягу системної пам'яті. Він використовує схему розділів GPT, підтримуючи диски обсягом понад 2 ТБ, та може легко оновлюватися через завантаження мікропрограми з сайту виробника. Доступ до UEFI здійснюється під час перезавантаження або увімкнення живлення натисканням відповідних клавіш. Меню UEFI дозволяє конфігурувати порядок завантаження, параметри безпеки та налаштування обладнання (рис. 11.27).

Рисунок 11.27 – Утиліта налаштування UEFI [13]

Довірений платформний модуль (TPM) – це чіп безпеки, вбудований у материнську плату сервера, призначений для захищеного зберігання ключів шифрування, сертифікатів та паролів. TPM відіграє вирішальну роль у вимірюванні цілісності процесу завантаження, гарантуючи відсутність несанкціонованих змін у мікропрограмі, завантажувачі або операційній системі. Він працює в поєднанні з BitLocker, який шифрує диски сервера, використовуючи TPM для зберігання ключа шифрування та розблокування його лише після успішної перевірки цілісності (рис. 11.28). Це забезпечує надійний захист даних від крадіжки та втручання [13].

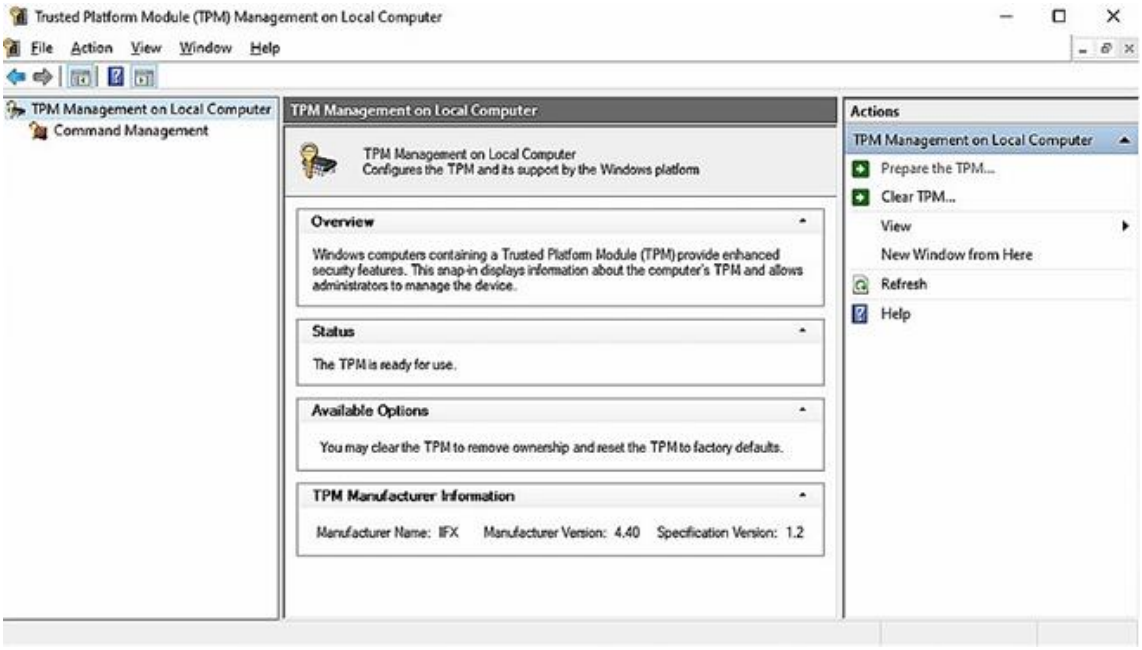

Рисунок 11.28 – Консоль керування TPM [13]

Для коректного запуску сервера проводиться автоматичний діагностичний тест POST (Power-On Self-Test). POST перевіряє процесор, пам’ять, диски та інші пристрої на наявність помилок, повідомляючи про проблеми через звукові коди або повідомлення на екрані. Особлива увага приділяється таким компонентам, як процесори та відеокарти, оскільки у разі їх несправності сервер не завантажиться. Оскільки різні виробники BIOS/UEFI використовують різні звукові коди, ознайомлення з ними є корисним для діагностики апаратних збоїв [13].

Після проходження POST керування передається першому завантажувальному пристрою. BIOS/UEFI сканує пристрій на наявність таблиці розділів (MBR або GPT), яка вказує розташування ОС. GPT є сучасним стандартом, що підтримує більші диски та підвищує надійність завдяки дублюванню таблиці розділів. UEFI використовує завантажувач, здатний читати розділи GPT. Залежно від версії Windows, завантажувачем може бути NTLDR (для старих версій) або BOOTMGR (від Windows Vista до Windows Server 2025).

Дані конфігурації завантаження (BCD) – це база даних, що зберігає налаштування

завантаження ОС. BCD керується за допомогою інструменту командного рядка bcdedit.exe або графічних засобів (рис. 11.29). Вона містить записи для кожного завантажувача та ОС, а також параметри налагодження та відновлення. BCD забезпечує стандартизований інтерфейс опцій завантаження, підвищуючи безпеку порівняно з попередньою системою boot.ini [13].

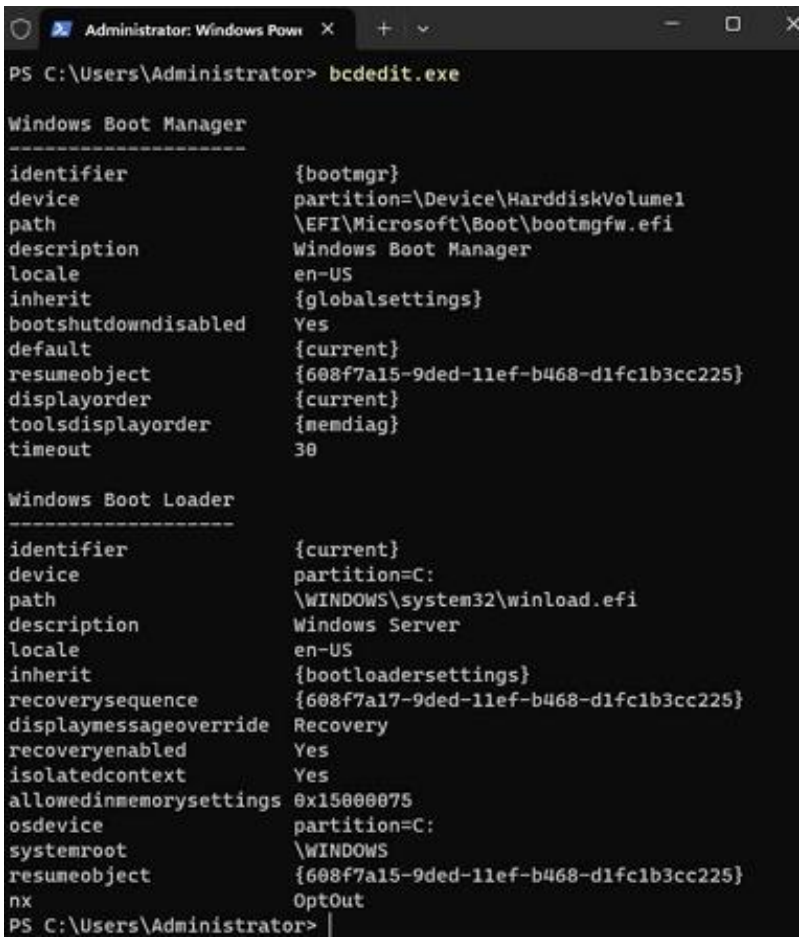

Рисунок 11.29 – Запуск bcdedit.exe у Windows Server 2025 [13]

У сценаріях із кількома завантаженнями (multiboot) можуть бути присутні як NTLDR, так і BOOTMGR. Проблеми з розбиттям дисків та сумісністю драйверів є поширеними під час інсталяції і вирішуються шляхом перевірки режиму завантаження, використання сумісних стилів розділів та завантаження актуальних драйверів.

Завантажувач (bootloader) – це програма, що ініціює запуск системи після POST, завантажуючи ядро ОС у пам’ять [18].

Завантажувальний сектор – це критична область на диску, що містить MBR або GPT. У системах BIOS із MBR сектор містить головний код завантаження та таблицю розділів. Системи UEFI з GPT використовують системний розділ EFI (ESP). Для сумісності UEFI може використовувати модуль підтримки сумісності (CSM) для імітації BIOS. Меню завантаження дозволяє вибирати між кількома встановленими ОС. У старіших версіях це керувалося файлом boot.ini, тоді як нові версії використовують базу даних BCD [18].

Безпечний режим (Safe Mode) є діагностичним інструментом, що запускає Windows лише з основними драйверами та службами. Доступ до нього у старіших версіях здійснювався через клавішу F8 [18].

У нових версіях, включаючи Windows Server 2025, використовуються розширені параметри запуску (Advanced startup options), доступні через утримання клавіші Shift під час перезавантаження. Після перезапуску з’являється екран розширених параметрів завантаження, де можна обрати безпечний режим (рис. 11.30).

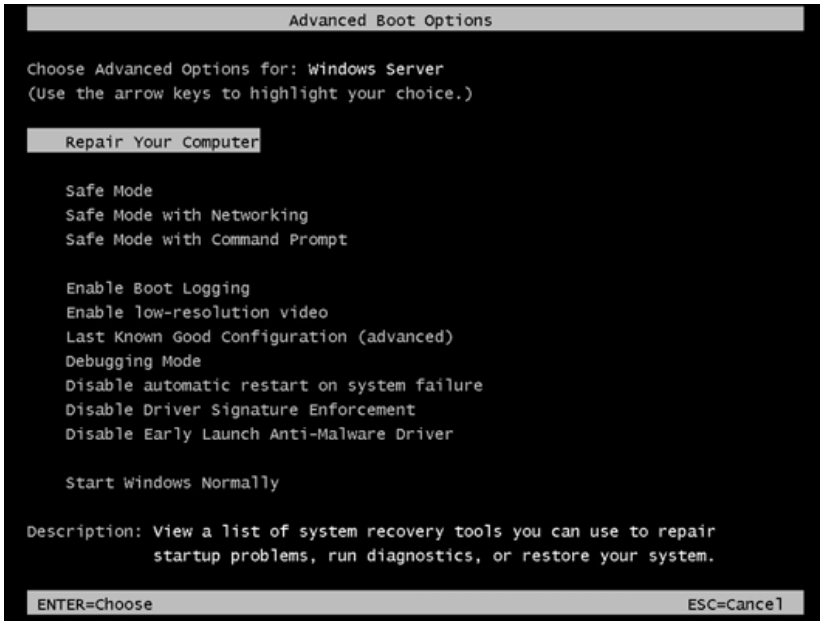

Рисунок 11.30 – Розширені параметри завантаження [13]

Під час підготовки до інсталяції Windows Server 2025 необхідно враховувати вимоги до файлової системи. Системні диски повинні бути відформатовані в NTFS. Для керування розділами використовується утиліта «Керування дисками» (Disk Management), яка дозволяє форматувати диски, змінювати розмір томів (стискати або розширювати) та перевіряти сумісність. Забезпечення правильної конфігурації дисків та формату NTFS мінімізує помилки під час інсталяції.

У цьому розділі було досліджено елементи процесу завантаження Windows, включаючи BIOS, UEFI, TPM, POST, MBR, BCD, завантажувачі та безпечний режим. Наступний розділ зосередиться на безперервності бізнесу та стратегії її підтримки.

Варіанти встановлення Windows Server 2025

При розгортанні Windows Server вибір відповідного варіанта встановлення є критично важливим етапом для задоволення специфічних потреб інфраструктури. Windows Server надає різноманітні варіанти інсталяції, кожен з яких відповідає різним вимогам щодо дискового простору, використання пам’яті, функціональних можливостей та графічних інтерфейсів. Ці опції також безпосередньо впливають на безпеку, продуктивність, керування та сумісність системи.

Під час інсталяції Windows Server 2025 важливо враховувати специфічні ролі та обов’язки, які виконуватиме сервер. Це передбачення може суттєво вплинути на вибір конфігурації під час інсталяції, включаючи специфікації апаратного забезпечення та вибір служб. Оцінка робочих навантажень передбачає визначення того, чи буде сервер переважно обробляти операції читання, запису або збалансоване поєднання обох. Наприклад, файловий сервер, який переважно надає файли кільком клієнтам, може виграти від вищих швидкостей диска та більшого обсягу оперативної пам’яті для задоволення високих вимог до читання. Натомість сервер баз даних, що обробляє інтенсивні операції запису, може потребувати оптимізованих рішень для зберігання даних та потенційно потужніших обчислювальних ресурсів [19].

Вимоги до пам’яті також варіюються залежно від ролі сервера. Сервери віртуалізації зазвичай потребують більше RAM для ефективного керування кількома віртуальними машинами. Забезпечення адекватного обсягу пам’яті сприяє підтримці продуктивності та швидкодії під навантаженням. Вибір компонентів та служб є критичним: для веб-сервера пріоритетом можуть бути мережеві інтерфейсні карти (NIC) для високошвидкісного з’єднання, тоді як для сервера баз даних акцент робиться на швидких дисках великої ємності для ефективної обробки транзакцій даних. Також важливим є планування майбутнього зростання, що передбачає врахування необхідності обслуговування більшої кількості користувачів або обробки збільшених обсягів даних. Ці фактори слід враховувати при виборі апаратного забезпечення для уникнення потенційних вузьких місць у міру зростання вимог.

Інтеграція цих міркувань у планування інсталяції дозволяє адаптувати середовище Windows Server 2025 для ефективного задоволення організаційних потреб.

Перед початком інсталяції Windows Server 2025 необхідно виконати ретельну перевірку сумісності ресурсів для забезпечення відповідності апаратного забезпечення необхідним вимогам для оптимальної продуктивності. Цей крок дозволяє заощадити час та запобігти проблемам під час та після процесу інсталяції. Ключові пункти включають перевірку системних вимог: процесор повинен бути сумісним з Windows Server 2025 (зазвичай мінімум 1,4 ГГц, 64-бітний процесор), при цьому рекомендується багатоядерний процесор для кращої продуктивності. Мінімальна вимога до ОЗП становить 2 ГБ, проте для кращої продуктивності та обробки важчих навантажень рекомендується 4 ГБ або більше [15].

Також перевіряється доступність ресурсів, зокрема дискового простору. Необхідно забезпечити достатньо місця для інсталяції, а також для майбутніх оновлень та програм (рекомендується мінімум 32 ГБ вільного простору). Оцінюються мережеві ресурси, зокрема пропускна здатність та з'єднання, особливо при розгортанні у хмарному або гібридному середовищі. Встановлення Windows Server на обладнання з недостатньою потужністю може призвести до проблем із продуктивністю, що вплине на роботу сервера та запущених на ньому програм. Додатково перевіряється сумісність існуючих програм з новою версією операційної системи. Проведення цих перевірок перед інсталяцією створює основу для успішного розгортання Windows Server 2025 [15].

При встановленні Windows Server 2025 пропонуються три різні варіанти, кожен з яких має унікальні переваги та обмеження.

Опція Desktop Experience надає повний графічний інтерфейс користувача (GUI) разом з усіма інструментами та функціональними можливостями Windows Server 2025. Хоча це забезпечує комплексний користувацький досвід, цей варіант вимагає більше апаратних ресурсів і може становити вищий ризик безпеки порівняно з іншими.

Опція Server Core, рекомендована Microsoft за свою ефективність, є варіантом мінімальної інсталяції, який виключає GUI, зосереджуючись на основних функціях сервера. Вона споживає менше ресурсів та має зменшену площу атаки. Керування здійснюється локально через Windows PowerShell або віддалено за допомогою Server Manager.

Третім варіантом є Nano Server – вдосконалена версія Server Core, розроблена для ще більшої легкості та ефективності. Вона підтримує лише 64-бітні додатки та не має можливості локального входу, вимагаючи керування через віддалені інструменти, такі як Windows Admin Center або Windows PowerShell. Nano Server особливо підходить для хмарних середовищ або контейнеризованих додатків.

При розгортанні Windows Server 2025 забезпечення надійного мережевого підключення є критичним для успішного приєднання до домену. Процедура усунення несправностей включає перевірку конфігурації IP (підтвердження коректності статичної адреси або отримання адреси від DHCP) та налаштувань DNS, оскільки DNS є життєво важливим для пошуку контролерів домену.

Необхідно перевірити конфігурації брандмауера, щоб переконатися, що відповідні порти відкриті (53 для DNS, 88 для Kerberos, 389 для LDAP). Використання команди ping дозволяє перевірити зв'язок з контролером домену. Також здійснюється огляд налаштувань мережевого адаптера та аналіз журналів подій на предмет помилок, пов'язаних з мережею.

Питання активації та ліцензування є важливими при інсталяції Windows Server 2025, особливо в Azure або гібридних середовищах. Належна активація гарантує автентичність ОС та можливість отримання оновлень. Поширеною проблемою є збій активації через проблеми зі з'єднанням. У хмарних налаштуваннях необхідно забезпечити стабільне інтернет-з'єднання з серверами активації Microsoft. Якщо активація не вдається, доцільним є використання інструменту командного рядка slmgr, наприклад, виконання команди slmgr /ato для спроби ручної активації [13].

Проблеми ліцензування також можуть виникати при використанні ключів корпоративного ліцензування. Для гібридних розгортань слід розглянути перевагу гібридного використання Azure (Azure Hybrid Benefit), яка дозволяє застосовувати існуючі ліцензії Windows Server до віртуальних машин Azure. Вирішення цих питань на ранніх етапах є

життєво важливим для безперебійного розгортання та відповідності вимогам. Далі будуть розглянуті різні методи розгортання Windows Server 2025 [13].

Загалом, існує кілька методологічних підходів до інсталяції Windows Server 2025, кожен з яких адаптований до конкретних сценаріїв експлуатації.

Основним методом є «чиста» установка (Clean install), яка передбачає розгортання нового екземпляра операційної системи з повним видаленням попередніх даних та конфігурацій. Цей підхід є оптимальним для нових розгортань або у випадках, коли необхідно розпочати роботу з «чистого аркуша» на новому чи існуючому жорсткому диску. Процес ініціюється завантаженням сервера із зовнішнього носія (DVD, USB) або через мережу, після чого інсталяційні файли завантажуються в оперативну пам'ять. У ході налаштування здійснюється вибір мовних параметрів, прийняття ліцензійних умов та вибір редакції операційної системи, наприклад, Windows Server 2025 Datacenter (Desktop Experience), як показано на рисунку 11.32.

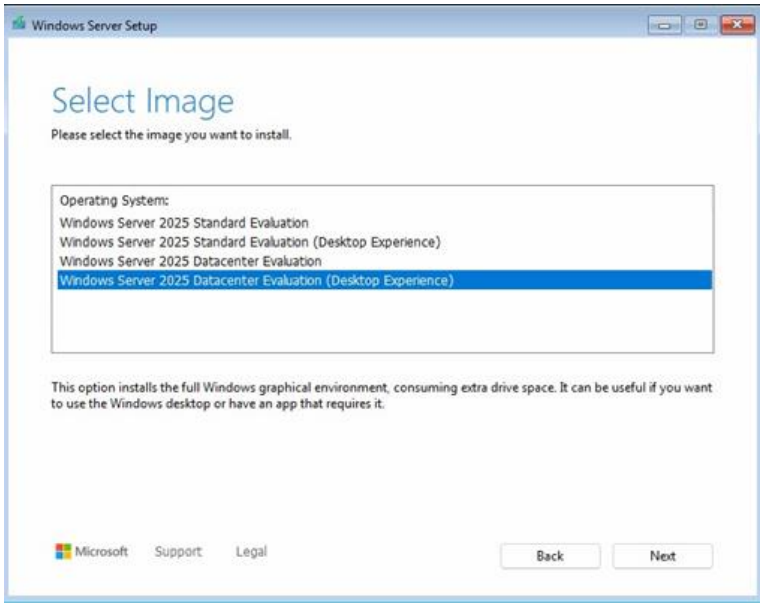

Рисунок 11.32 – Вибір образу операційної системи для інсталяції [13]

Критичним етапом є вибір цільового диска або розділу для інсталяції, де підтверджується видалення існуючих даних, що ілюструється на рисунку 11.33. Після завершення копіювання файлів та налаштування компонентів система перезавантажується для встановлення пароля адміністратора та початкового входу в систему. Важливим аспектом архітектури є використання Windows Installer – інтерфейсу прикладного програмування (API) для керування встановленням та обслуговуванням програмного забезпечення.

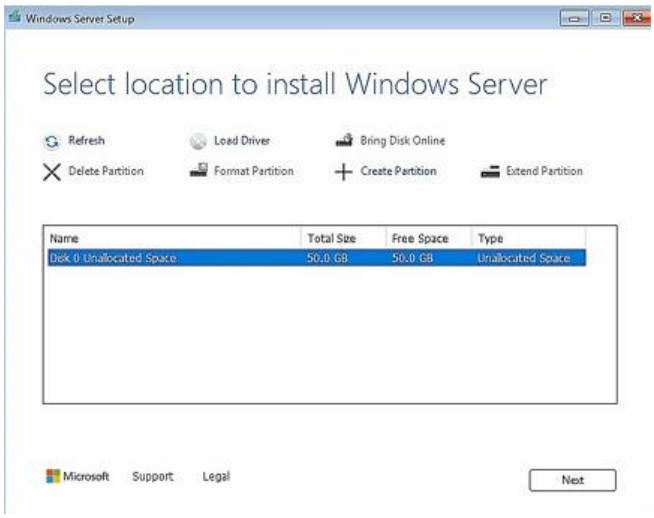

Рисунок 11.33 – Вибір диска або розділу для встановлення Windows Server 2025 [13]

Для оптимізації процесів у корпоративних середовищах застосовується розгортання за допомогою Microsoft Deployment Toolkit (MDT). Цей метод дозволяє автоматизувати інсталяцію, мінімізуючи необхідність ручного втручання, що є критично важливим при масовому розгортанні серверів. Незважаючи на те, що службу розгортання Windows (WDS) визнано застарілою, використання комплекту оцінки та розгортання Windows (Windows ADK) у поєднанні з MDT залишається актуальним стандартом. Ключовим елементом автоматичної установки є файл відповідей у форматі XML, який містить попередньо визначені параметри конфігурації [13].

Створення такого файлу здійснюється за допомогою диспетчера системних образів Windows (Windows SIM) або безпосередньо в середовищі MDT. Процедура передбачає інсталяцію Windows ADK та середовища попереднього встановлення (Windows PE), після чого налаштовується спільний ресурс розгортання через майстер налаштування, як зображено на рисунку 3.10. Після імпорту файлів операційної системи створюється послідовність завдань, яка визначає логіку інсталяції. Сервер завантажується за допомогою образу Windows PE, підключається до спільного ресурсу та виконує розгортання згідно з визначеним сценарієм, що демонструється на рисунку 11.34.

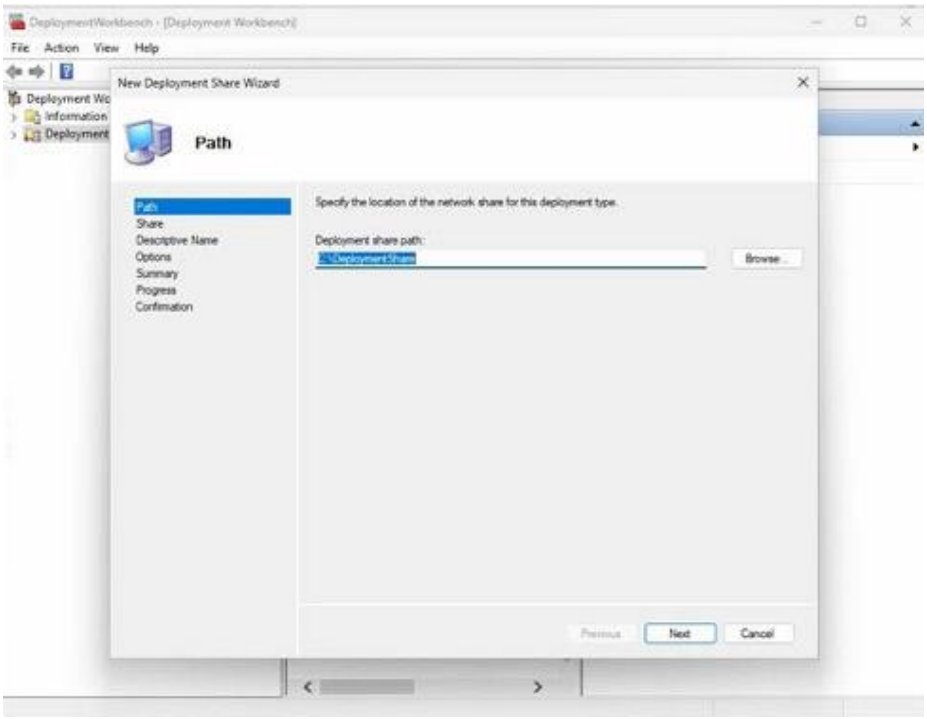

Рисунок 11.34 – Майстер створення нового спільного ресурсу розгортання [13]

Рисунок 11.35 – Розгортання Windows Server 2025 через MDT [13]

Альтернативним підходом є оновлення на місці (In-place upgrade), яке дозволяє модернізувати існуючу операційну систему до версії Windows Server 2025 зі збереженням налаштувань користувача, встановлених додатків та даних. Цей метод підтримується для переходу з версій Windows Server 2012 R2, 2016, 2019 або 2022. Перед початком процедури обов'язковим є виконання резервного копіювання стану системи. Процес оновлення ініціюється із середовища працюючої ОС шляхом запуску файлу налаштування з інсталяційного носія. Після вибору опції збереження файлів та налаштувань система проводить перевірку сумісності та наявності вільного дискового простору. Запуск процесу оновлення, показаний на рисунку 11.36, призводить до заміни системних файлів на нові версії з подальшим перезавантаженням сервера.

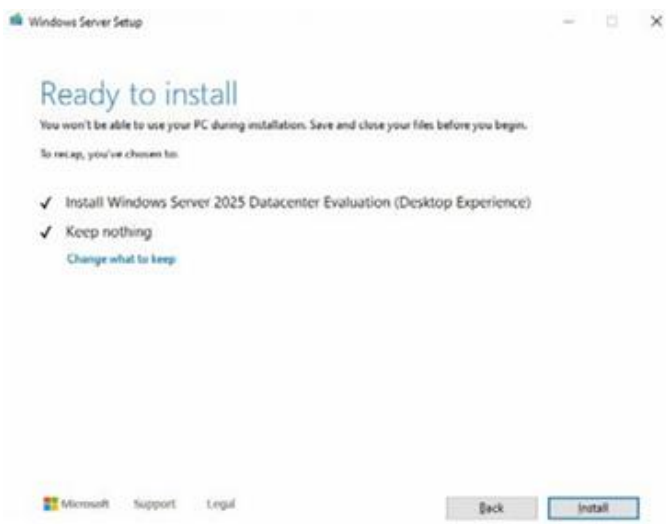

Рисунок 11.36 – Готовність до виконання оновлення на місці [13]

Окремим важливим процесом є міграція сервера, яка передбачає перенесення ролей, функцій, програм та мережевих служб зі старого сервера на новий, що працює під управлінням Windows Server 2025. Цей процес відрізняється від оновлення тим, що виконується між різними фізичними або віртуальними машинами. Для реалізації міграції використовуються інструменти міграції Windows Server (WSMT) та командлети PowerShell. Прикладом може слугувати міграція DHCP-сервера, де на вихідному сервері виконується експорт конфігурації у XML-файл за допомогою команди Export-DhcpServer. Після встановлення ролі DHCP на новому сервері здійснюється імпорт налаштувань за допомогою командлета Import-DhcpServer, як проілюстровано на рисунку 11.37. Це забезпечує безперервність надання мережевих послуг при зміні апаратної або програмної платформи.

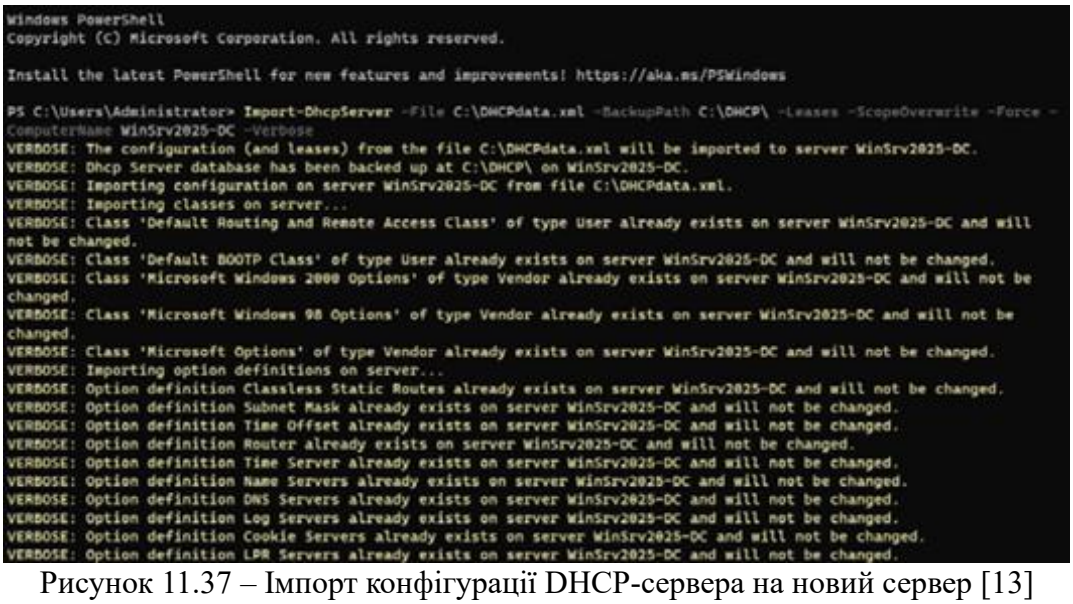

Рисунок 11.37 – Імпорт конфігурації DHCP-сервера на новий сервер [13]

В умовах сучасної цифровізації набуває поширення розгортання в хмарному середовищі Microsoft Azure. Цей метод дозволяє використовувати Windows Server 2025 як віртуальну машину, забезпечуючи масштабованість та надійність без необхідності у фізичному обладнанні. Процес передбачає створення облікового запису Azure та налаштування віртуальної машини через веб-портал. Під час конфігурації визначаються група ресурсів, регіон розташування та образ операційної системи, зокрема Windows Server 2025 Datacenter – Gen2, що показано на рисунку 11.38. Після задання параметрів дисків та мережі ініціюється розгортання, по завершенні якого доступ до сервера здійснюється через протокол віддаленого робочого столу (RDP). Цей підхід є оптимальним для гібридних інфраструктур та тестування нових можливостей ОС [13].

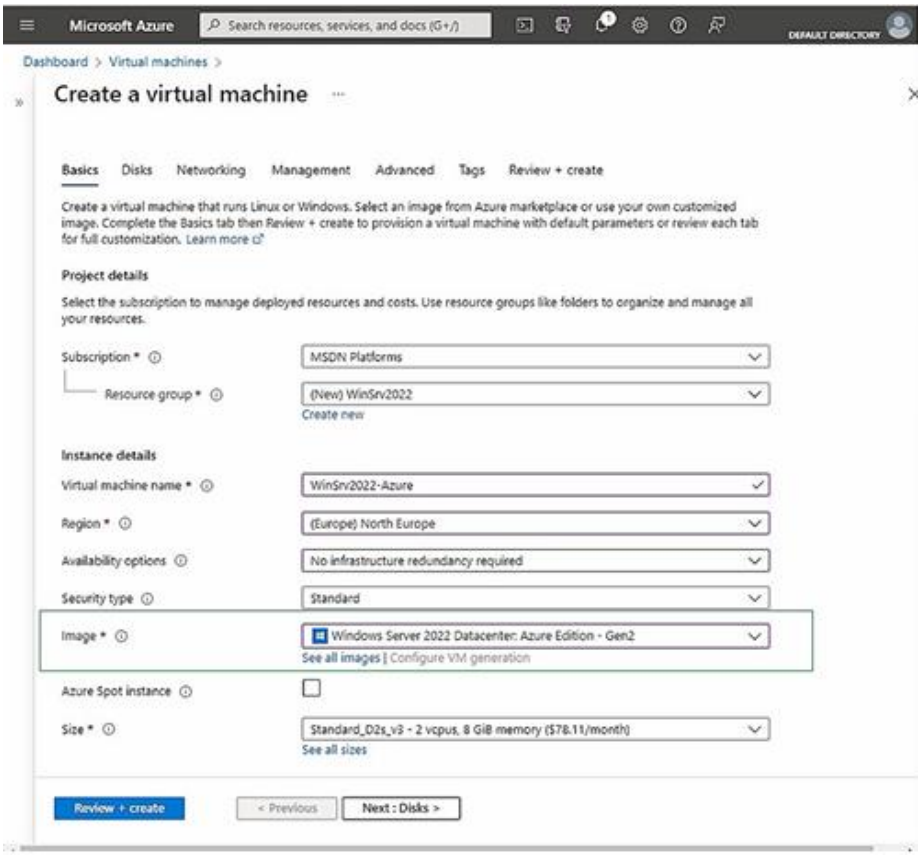

Рисунок 11.38 – Налаштування віртуальної машини з Windows Server 2025 в Azure [13]

Ролі та компоненти ОС

Перед призначенням ролей серверу необхідно чітко визначити його передбачувану функцію в IT-інфраструктурі організації. Цей розділ забезпечує всебічний огляд різноманітних ролей, служб ролей та компонентів, доступних у Windows Server 2025, що сприяє прийняттю обґрунтованих рішень щодо конфігурації сервера.

Роль сервера визначає основну функцію, яку сервер виконує в мережі. Наприклад, якщо основною метою сервера є зберігання та керування спільними файлами, для виконання цього завдання інсталюється роль «Файлові служби та служби зберігання» (File and Storage Services). Аналогічно, сервер, призначений для розміщення веб-додатків, використовуватиме роль «Веб-сервер (IIS)» для безпечної обробки запитів HTTP, забезпечуючи важливу платформу для інтернет- та інтранет-сервісів. Сервери, що уможливлюють безпечний віддалений доступ, реалізують роль «Віддалений доступ» (Remote Access), яка полегшує рішення для підключення, такі як віртуальні приватні мережі (VPN) та DirectAccess, дозволяючи користувачам безпечно отримувати доступ до мережевих ресурсів з віддалених локацій. У більшості випадків призначення однієї ролі кожному серверу є оптимальним, оскільки це забезпечує оптимізовану продуктивність та спрощує керування сервером. Однак існують сценарії, коли на одному сервері може бути розгорнуто кілька ролей. У таких випадках необхідне ретельне планування для збалансування апаратних ресурсів відповідно

до вимог конкретних ролей, забезпечення сумісності та запобігання потенційним конфліктам або вузьким місцям у продуктивності. Такий модульний підхід дозволяє Windows Server 2025 служити для низки цілей в організації, де кожна роль робить внесок у надійну, чуйну та безпечну мережеву інфраструктуру [13].

Окрім базових ролей, Windows Server 2025 пропонує служби ролей – додаткові компоненти, що покращують або розширюють функціональність ролі сервера. Ці служби дозволяють адміністраторам адаптувати можливості сервера до конкретних потреб. Наприклад, увімкнення віддаленого друку через Інтернет вимагає не лише встановлення ролі служб друку та документів (Print and Document Services), але й додавання служби ролі «Інтернет-друк». Такий багаторівневий підхід дозволяє налаштувати функціональність сервера для відповідності точним операційним вимогам, забезпечуючи гнучкість у тому, як сервер підтримує потреби організації [13].

Windows Server 2025 включає ряд вбудованих ролей та компонентів, розроблених для підтримки критичних потреб інфраструктури без покладання на додаткові додатки. Вибір для висвітлення трьох конкретних ролей – служб сертифікації Active Directory (AD CS), служб керування правами (RMS) та сервера політик мережі (NPS) – ґрунтується на їхній актуальності для фундаментальних аспектів керування Windows Server: безпеки, контролю доступу та захисту даних. Ці ролі забезпечують суттєві інфраструктурні можливості, на які покладаються багато організацій, незалежно від додаткових додатків, таких як Exchange або SQL Server.

Служби сертифікації Active Directory (AD CS) забезпечують масштабований, безпечний метод видачі та керування цифровими сертифікатами в організації. Це є фундаментальним для підтримки безпечної комунікації, цілісності даних та автентифікації користувачів. Роль відіграє вирішальне значення в середовищах, що пріоритезують безпеку, уможливлюючи виконання таких завдань, як SSL/TLS для веб-сайтів та автентифікація користувачів і пристроїв.

Служби керування правами (RMS) є життєво важливими для захисту інформації та убезпечення конфіденційних документів і комунікацій шляхом забезпечення дотримання обмежень доступу та використання. Це гарантує, що лише авторизовані користувачі можуть взаємодіяти із захищеним вмістом, що є особливо критичним у галузях, які працюють з чутливими або регульованими даними.

Сервер політик мережі (NPS) функціонує як сервер RADIUS, підтримуючи централізовану автентифікацію доступу до мережі, авторизацію та облік. Ця можливість є безцінною для керування безпечним доступом до мережі, особливо в середовищах, де важлива інтеграція кількох сайтів та хмари, полегшуючи безпечні з’єднання для VPN, бездротових мереж та інших рішень віддаленого доступу.

Хоча ці ролі є критичними, Windows Server 2025 також включає кілька інших цінних вбудованих функцій, які потребують дослідження. До них належать «Файлові служби та служби зберігання», які є невід’ємною частиною централізованого спільного доступу до файлів, керування зберіганням та дедуплікації даних, вирішуючи основні потреби в мережевих середовищах. Роль Hyper-V є важливою для організацій, що використовують віртуалізацію, оптимізуючи використання серверів та забезпечуючи ізольовані віртуальні середовища для різних додатків.

Ролі DNS та DHCP є фундаментальними для мережевої інфраструктури, забезпечуючи розпізнавання доменних імен та керування IP-адресами (IPAM). Служби оновлення Windows Server (WSUS) є критичними для керування виправленнями, гарантуючи, що сервери та підключені пристрої отримують своєчасні оновлення для підтримання безпеки та відповідності вимогам. Розширення огляду на ширший вибір цих ролей забезпечує більш повне уявлення про вбудовані можливості Windows Server, оснащуючи адміністраторів міцною основою для керування безпекою мережі, відповідністю та доступністю. Це розуміння закладає підґрунтя для розширення функціональних можливостей сервера за допомогою додаткових додатків, підкреслюючи важливість вбудованих функцій у середовищах Windows Server [13].

На додаток до ролей та служб ролей, компоненти сервера є допоміжними елементами,

що підтримують або покращують конкретні функції в серверному середовищі. Наприклад, встановлення компонента .NET Framework 3.5 може бути необхідним для запуску певних програм або служб. Натомість компонент IPAM надає розширені можливості керування для ролей DHCP та DNS. Такі компоненти, як WINS, можуть бути вирішальними в середовищах, де необхідне розпізнавання імен NetBIOS у кількох підмережах. Шляхом ретельного вибору та встановлення відповідних компонентів забезпечується повна оснащеність сервера для ефективного виконання призначених завдань, що сприяє загальній стабільності та продуктивності IT-інфраструктури. Підсумовуючи, розуміння та стратегічне налаштування ролей, служб ролей та компонентів у Windows Server 2025 є ключовим для оптимізації продуктивності сервера та задоволення унікальних потреб IT-середовища організації.

Диспетчер серверів (Server Manager) є основним інструментом для додавання, налаштування та керування ролями серверів у Windows Server 2025. Вперше представлений у Windows Server 2008, цей інструмент постійно вдосконалювався, пропонуючи оптимізований та інтуїтивно зрозумілий інтерфейс, що спрощує адміністрування сервера. Незалежно від того, чи виконується робота з локальним сервером, чи здійснюється керування віддаленими серверами, Server Manager дозволяє ефективно встановлювати та контролювати ролі серверів. Інтерфейс розділено на дві основні секції: панель області, яка відображає всі встановлені ролі, та панель деталей, яка надає вичерпну інформацію та опції керування для кожної обраної ролі. Ця центральна консоль не лише допомагає у моніторингу стану та продуктивності сервера, але й дозволяє легко отримувати доступ до інструментів та налаштувань, специфічних для ролі (рис. 11.39).

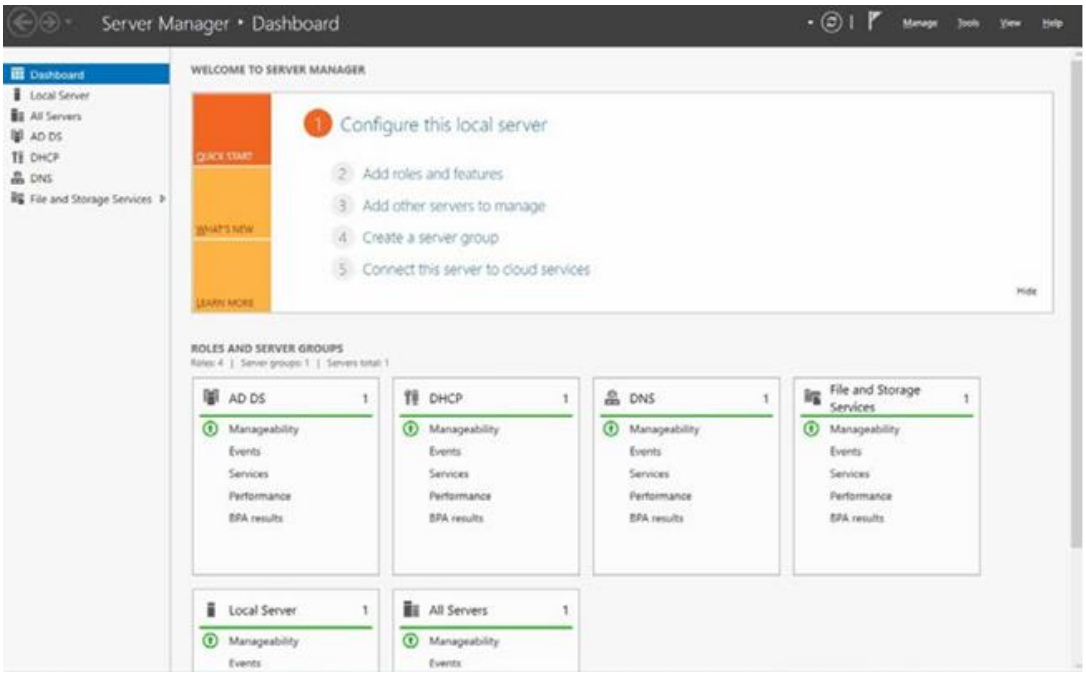

Рисунок 11.39 – Інтерфейс Диспетчера серверів у Windows Server 2025 [13]

Налаштування різних видів доступу

Ефективне адміністрування серверної інфраструктури неможливе без розуміння та правильної конфігурації механізмів доступу. У середовищі Windows Server 2025 реалізовано багаторівневу модель керування, яка передбачає перехід від локальної взаємодії до віддалених, автоматизованих та веб-орієнтованих інтерфейсів. Вибір методу доступу визначається політикою безпеки, архітектурою мережі (локальна, гібридна, хмарна) та специфікою розгорнутих ролей.

Найпоширенішим методом графічного керування сервером є використання протоколу RDP (Remote Desktop Protocol). В операційних системах сімейства Windows Server за замовчуванням дозволяється два одночасні адміністративні сеанси без необхідності розгортання ролі сервера терміналів та придбання клієнтських ліцензій (CAL). Доступ здійснюється через порт TCP 3389 [15].

Налаштування здійснюється через властивості системи або за допомогою командлетів PowerShell (Set-ItemProperty), де модифікується реєстр для дозволу підключень. Слід зауважити, що використання RDP для адміністрування контролерів домену вимагає підвищених заходів безпеки, зокрема використання виділених адміністративних робочих станцій (PAW – Privileged Access Workstations) [15].

Сучасним стандартом керування Windows Server 2025 є Windows Admin Center – інструмент, що розгортається локально або на шлюзовому сервері. Архітектурно WAC працює через веб-браузер, використовуючи протокол HTTPS (порт 443 за замовчуванням) для зв'язку з керуючим вузлом, який, у свою чергу, комунікує з керованими серверами через PowerShell Remoting та WMI (Windows Management Instrumentation) поверх WinRM [13].

WAC консолідує більшість адміністративних завдань (керування сертифікатами, моніторинг подій, налаштування мережі, керування оновленнями) в єдиному інтерфейсі. Особливістю WAC є його здатність керувати як локальними серверами, так і віртуальними машинами в Azure, забезпечуючи «єдине вікно» для гібридних інфраструктур. Цей інструмент не вимагає встановлення агентів на цільові сервери, що спрощує його впровадження.

Основним інструментом автоматизації та віддаленого керування серверами, зокрема у конфігурації Server Core, є технологія PowerShell Remoting. Вона базується на протоколі WS-Management та службі WinRM, яка за замовчуванням використовує порти 5985 (HTTP) та 5986 (HTTPS) для забезпечення каналу зв'язку між керуючою станцією та серверами.

Взаємодія реалізується двома методами: через інтерактивні сесії за допомогою командлета Enter-PSSession для керування окремим сервером у реальному часі, або шляхом масового виконання команд через Invoke-Command для одночасної обробки скриптів на групі вузлів. Активація функціоналу здійснюється командою Enable-PSRemoting з можливістю додаткового налаштування безпеки через шифрування та обмеження списків довірених хостів.

RSAT (Remote Server Administration Tools) представляє собою набір класичних оснасток MMC (Microsoft Management Console) та інструментів командного рядка, які встановлюються на клієнтську робочу станцію адміністратора (наприклад, Windows 11). Цей підхід дозволяє керувати ролями сервера (DNS, DHCP, Active Directory) без необхідності прямого входу на консоль сервера через RDP [13].

В контексті адміністрування Windows Server 2025 важливим аспектом є реалізація принципу найменших привілеїв. Технологія JEA (Just Enough Administration) є надбудовою над PowerShell Remoting, яка дозволяє делегувати права адміністрування без надання повних прав локального адміністратора.

JEA функціонує шляхом створення віртуальних облікових записів та файлів конфігурації сесій, де чітко визначено, які командлети, параметри та модулі доступні конкретному користувачеві. Це дозволяє, наприклад, надати операторам служби підтримки право лише перезапускати службу DNS, забороняючи будь-які інші дії в системі. Впровадження JEA є критичним кроком для захисту від внутрішніх загроз та мінімізації наслідків компрометації облікових записів [13].

Механізми віддаленого управління

Роль «Віддалений доступ» у Windows Server 2025 є комплексним рішенням, що інтегрує технології для забезпечення безпечного підключення до корпоративних ресурсів. До її складу входять DirectAccess, який гарантує безперебійне з'єднання через тунелювання IPv6 поверх IPv4 та шифрування IPsec без необхідності використання традиційного VPN; служба маршрутизації та віддаленого доступу (RRAS), що підтримує маршрутизацію трафіку та створення захищених каналів між підмережами; а також Web Application Proxy, що діє як зворотний проксі-сервер для публікації внутрішніх веб-додатків із використанням автентифікації AD FS. Налаштування цих компонентів розпочинається з додавання відповідної ролі до сервера (рис. 11.40).

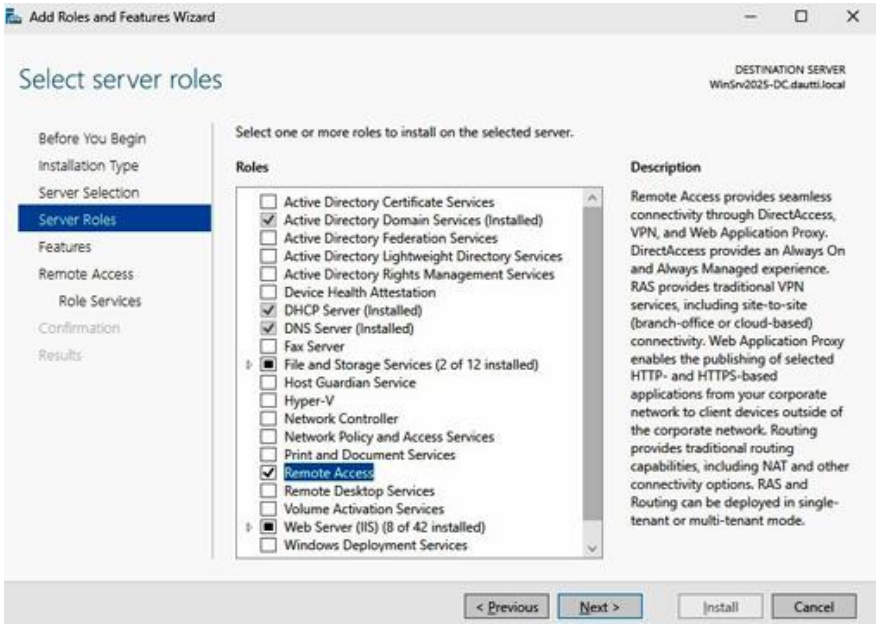

Рисунок 11.40 – Додавання ролі віддаленого доступу в Windows Server 2025 [13]

Механізм «Віддалений помічник» спроектовано для надання технічної підтримки в режимі реального часу, дозволяючи адміністратору (ініціатору) переглядати та керувати робочим столом користувача (запрошеного). Процес взаємодії регламентується суворими протоколами безпеки: сеанс ініціюється запитом користувача, який надає дозвіл на доступ, що забезпечує контроль над процедурою діагностики та усунення несправностей без фізичної присутності фахівця. Активація функції здійснюється через майстер додавання ролей та компонентів.

Інструменти віддаленого адміністрування сервера (RSAT) дозволяють здійснювати централізоване керування ролями та компонентами серверів Windows Server 2025 із клієнтських робочих станцій під управлінням Windows 10 або 11. Цей набір інструментів включає як графічні інтерфейси, так і засоби командного рядка, забезпечуючи гнучкість адміністрування без необхідності прямого входу на консоль сервера. На відміну від сучасного веб-орієнтованого Windows Admin Center, RSAT представляє класичний підхід до керування інфраструктурою (рис. 11.41).

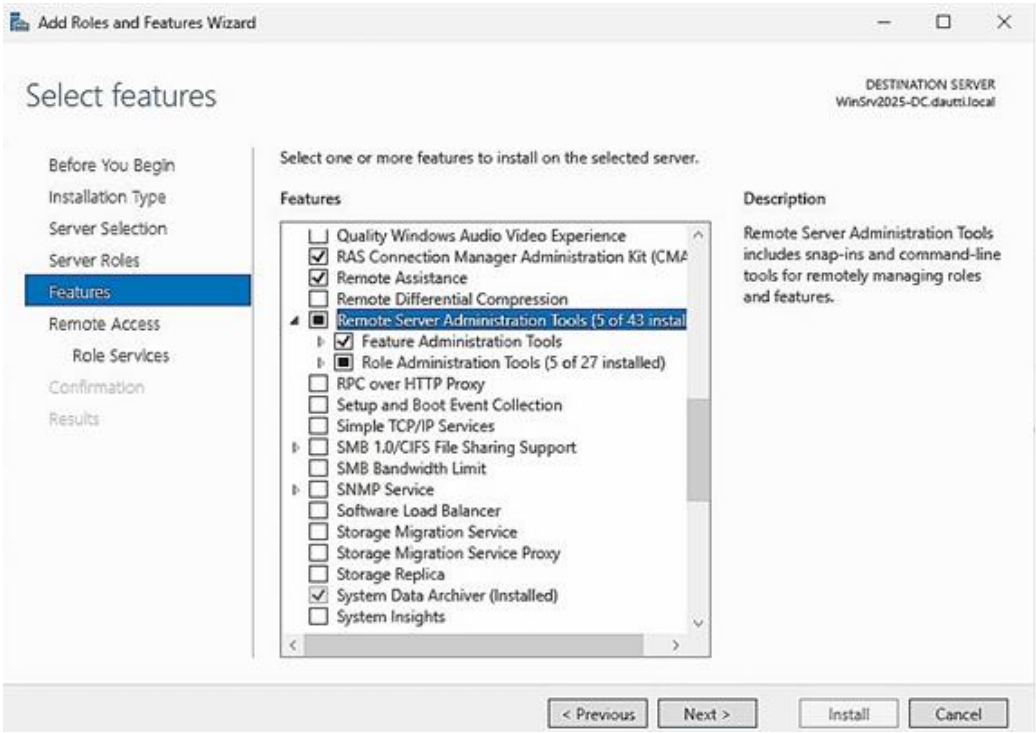

Рисунок 11.41 – Додавання компонента RSAT у Windows Server 2025 [13]

Служби віддалених робочих столів (RDS) забезпечують віртуалізацію сесій та додатків, дозволяючи користувачам працювати з графічним інтерфейсом сервера віддалено. Для функціонування розширеної інфраструктури (понад дві адміністративні сесії) вимагається розгортання сервера ліцензування для керування клієнтськими ліцензіями (RDS CALs) та налаштування шлюзу віддалених робочих столів (RDG). Шлюз виступає посередником, що інкапсулює RDP-трафік у HTTPS-пакети, забезпечуючи захищений доступ до внутрішніх ресурсів через Інтернет без прямого експонування серверів [13].

Технологія VPN реалізує захищений тунель для передачі даних через публічні мережі, використовуючи протоколи шифрування. У Windows Server 2025 підтримуються конфігурації віддаленого доступу (Remote-Access VPN) для підключення окремих клієнтів та з'єднання типу «сайт-сайт» для об'єднання географічно розподілених мереж. На відміну від DirectAccess, який забезпечує постійне автоматичне з'єднання для доменних комп'ютерів, класичний VPN є більш універсальним рішенням для різнорідних клієнтських пристроїв (рис. 11.42).

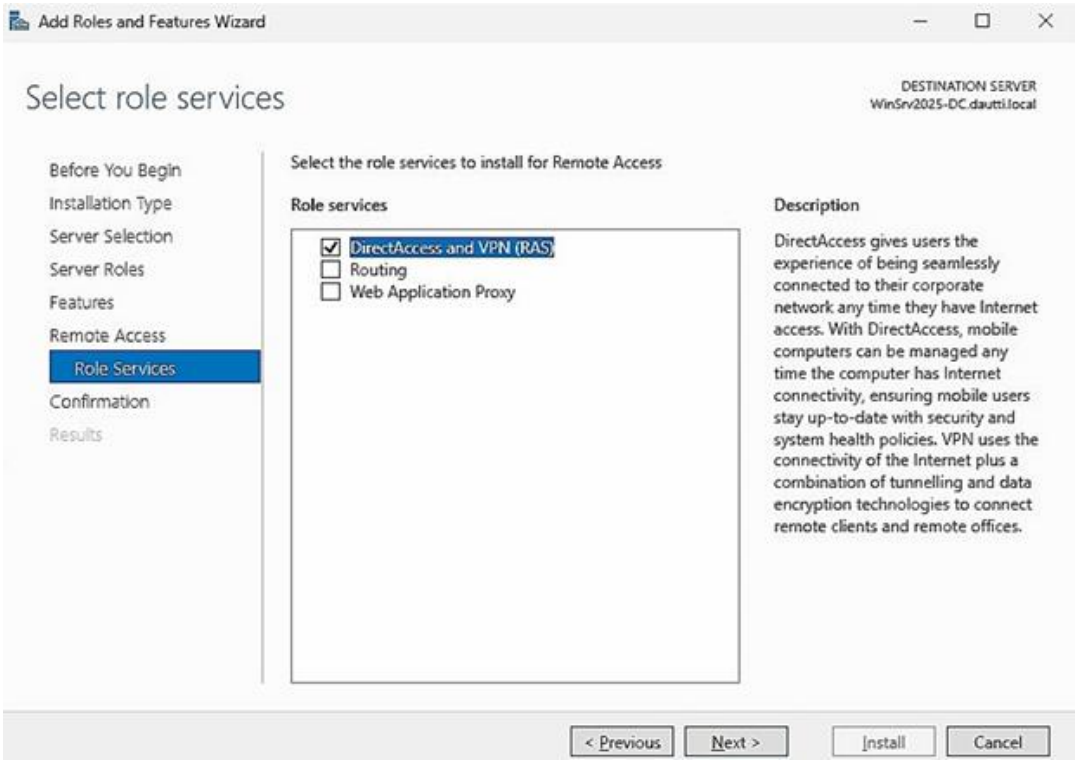

Рисунок 11.42 – Налаштування служб DirectAccess та VPN (RAS) [13]

Система Microsoft App-V дозволяє виконувати додатки без їх локальної інсталяції на кінцевих пристроях, використовуючи технологію потокової передачі з сервера. Це рішення ізолює додатки від операційної системи клієнта, мінімізуючи конфлікти сумісності (DLL hell) та спрощуючи процес оновлення програмного забезпечення. У сучасних сценаріях App-V часто інтегрується у хмарні середовища для підвищення масштабованості, а його впровадження вимагає використання пакету MDOP.

Для коректної маршрутизації трафіку в системах віддаленого доступу використовується механізм IP-сокетів, що поєднує IP-адресу вузла та номер порту (наприклад, 192.168.1.10:3389). Стандартний порт для RDP – 3389, проте для забезпечення одночасної роботи кількох сервісів або перенаправлення портів використовуються альтернативні значення (3390 і далі). Розуміння адресації на рівні сокетів є необхідним для налаштування правил брандмауера та коректної роботи служб RDS і App-V.

Резервне копіювання та архівація

Однією з першочергових відповідальностей в адмініструванні серверів є захист даних від втрати або пошкодження. Для реалізації цього завдання критично важливим є резервне копіювання, яке уможливлює дублювання даних на альтернативні локації або пристрої. Проте ефективність резервного копіювання безпосередньо залежить від здатності відновити

дані у разі необхідності. Процес відновлення передбачає вилучення даних із резервних копій та їх розгортання на сервері.

Існує кілька типів резервного копіювання, що задовольняють різні потреби залежно від частоти та обсягу робіт. Повне резервне копіювання створює повну копію всіх даних сервера; для відновлення потрібна лише остання повна копія, що забезпечує найпростіший процес реставрації.

Інкрементальне резервне копіювання зберігає лише дані, змінені з моменту останнього резервного копіювання будь-якого типу. Зазвичай воно виконується щодня, крім дня повного копіювання (часто п’ятниці), що пришвидшує процес бекапу, але вимагає наявності останньої повної копії та всіх наступних інкрементальних копій для відновлення, роблячи цей процес більш тривалим [15].

Диференційне резервне копіювання зберігає дані, модифіковані з моменту останнього повного копіювання. Для відновлення потрібні остання повна та найсвіжіша диференційна копії, що прискорює відновлення порівняно з інкрементальним методом, але може уповільнити процес створення копії [15].

Вибір носіїв для резервного копіювання залежить від важливості та обсягу даних. Доступні опції включають CD, DVD, знімні жорсткі диски, стрічкові накопичувачі, мережеві сховища (NAS) та мережі зберігання даних (SAN). Останнім часом організації все частіше впроваджують онлайн-сервіси резервного копіювання через їхню зручність, безпеку та економічну ефективність. Додатково широко застосовується схема ротації резервних копій «Дід-Батько-Син» (Grandfather-Father-Son, GFS), яка забезпечує структуровану та надійну стратегію: щоденні копії позначаються як «син», щотижневі – як «батько», а щомісячні – як «дід» [15].

У середовищі Windows Server 2025 ключовим інструментом захисту даних є компонент Windows Server Backup, активація якого здійснюється через консоль Server Manager. Процедура налаштування розпочинається з натискання комбінації клавіш Windows + R, введення команди servermanager.exe та підтвердження вводу. У консолі диспетчера серверів необхідно обрати пункт «Додати ролі та компоненти». Після проходження вступної сторінки та вибору типу встановлення «На основі ролей або компонентів» (Role-based or Feature-based Installation), обирається цільовий сервер із пулу. Оскільки додавання нових ролей не вимагається, цей крок пропускається. На сторінці вибору компонентів (Features) слід знайти у списку та відмітити пункт Windows Server Backup (рис. 11.43).

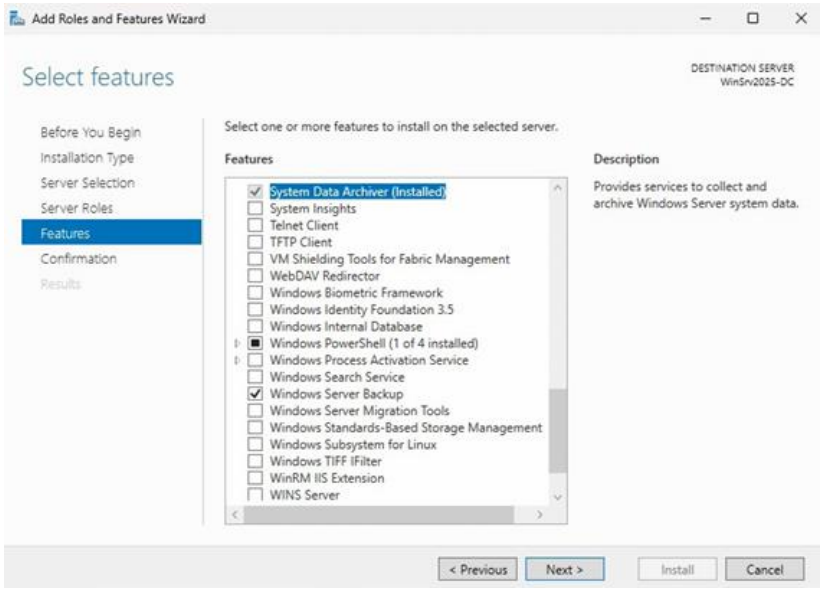

Рисунок 11.43 – Встановлення компонента Windows Server Backup [13]

Після підтвердження вибору натисканням кнопки «Install» на сторінці підтвердження розпочинається процес інсталяції, по завершенні якого роботу майстра слід завершити кнопкою «Close». Після встановлення Windows Server Backup система готова до виконання

подальших завдань, зокрема відновлення Active Directory.

Традиційні методи, такі як носії з одноразовим записом (WORM), цифрова аудіострічка (DAT) та технологія Travan (TDLT), стають менш поширеними через обмеження у масштабованості, швидкості та доступності за межами майданчика. Натомість сучасні рішення, такі як аварійне відновлення як послуга (DRaaS) та резервне копіювання як послуга (BaaS), набувають популярності завдяки можливостям безпечного, масштабованого та ефективного захисту даних у віддаленому режимі. Ці сервіси дозволяють організаціям зберігати копії у хмарних сховищах, гарантуючи збереження та доступність критичних даних навіть у разі катастроф. Використання DRaaS та BaaS дозволяє бізнесу вдосконалити стратегії бекапу, покращуючи можливості відновлення та загальну безперервність бізнес-процесів, а також масштабувати ресурси відповідно до зростання потреб без ускладнення локальної інфраструктури [13].

Аналіз системних журналів

Засіб перегляду подій (Event Viewer) є важливим інструментом для системних адміністраторів, спроектованим для сприяння усуненню несправностей в операційних системах Windows шляхом надання комплексного огляду подій, що відбуваються на серверах. Ця утиліта реєструє та відображає записи критичних дій та модифікацій, які впливають на програмне забезпечення, апаратне забезпечення або мережеві компоненти сервера. За допомогою засобу перегляду подій адміністратори мають можливість ефективно здійснювати моніторинг продуктивності та безпеки сервера, а також точно визначати першопричини різноманітних проблем. Шляхом аналізу зареєстрованих подій можна отримати цінну інформацію про операції системи та оперативно вирішувати проблеми, забезпечуючи оптимальну функціональність та стабільність сервера.

Засіб перегляду подій, інтерфейс якого представлено на рисунку 11.44, розроблено для полегшення діагностики різноманітних проблем шляхом надання детального звіту про події. Він пропонує для моніторингу п’ять різних типів журналів, кожен з яких слугує певній меті. Журнали програм (Application logs) фіксують події, згенеровані додатками, що виконуються на сервері. Журнали безпеки (Security logs) відстежують події, пов’язані з безпекою, такі як невдалі спроби входу або доступ до обмежених папок, що вимагає попереднього увімкнення аудиту. Журнали інсталяції (Setup logs) записують події, пов’язані зі встановленням та налаштуванням додатків, тоді як системні журнали (System logs) документують події, що стосуються функціонування компонентів ОС Windows [13].

Окремо виділяють журнали пересланих подій (Forwarded events logs), які містять події, зібрані з віддалених комп’ютерів, що потребує налаштування підписки. Варто зауважити, що необхідно коректно налаштувати мінімальний час зберігання журналів, оскільки за замовчуванням вони перезаписуються при досягненні максимального розміру файлу, що може призвести до втрати критичної інформації. Параметри зберігання коригуються через групову політику або редактор реєстру.

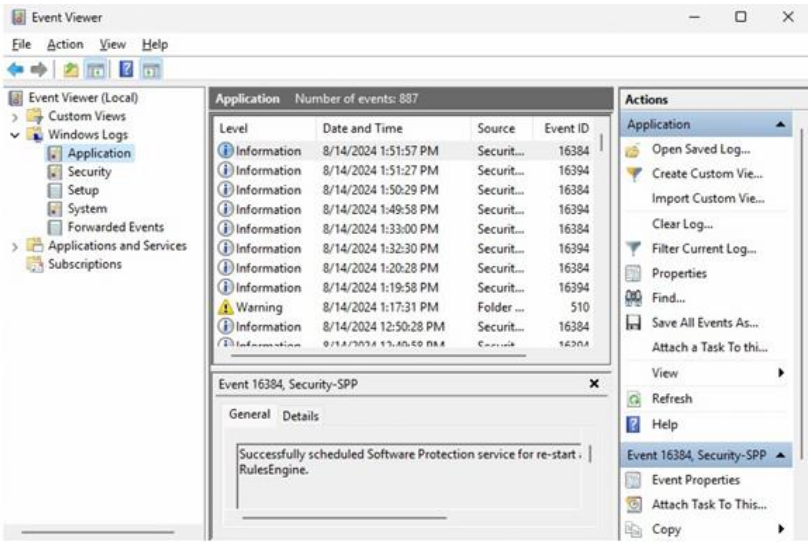

Рисунок 11.44 – Засіб перегляду подій (Event Viewer) [13]

Практичне застосування засобу перегляду подій передбачає виконання низки завдань, спрямованих на покращення нагляду за системою. Централізований моніторинг дозволяє збирати та переглядати події з декількох серверів в одній локації за допомогою налаштування підписки на події через Windows PowerShell або консоль Event Viewer. Це значно спрощує аналіз активності серверів.

Паралельно з цим, важливим аспектом є фільтрація журналів, яка необхідна для зосередження уваги на конкретних подіях, релевантних для діагностики. Застосування фільтрів базується на таких критеріях, як ім’я журналу, рівень події, дата, джерело, ID події або ключові слова, з можливістю збереження налаштувань як настроюваних подань. Крім того, адміністратори можуть змінювати розташування журналів за замовчуванням для звільнення місця на системному диску або покращення продуктивності, використовуючи редактор реєстру або команду wevtutil.

Реалізація централізованого моніторингу в Windows Server 2025 вимагає послідовного виконання конфігураційних дій. Процес розпочинається із запуску командного рядка від імені адміністратора та введення команди winrm quickconfig для налаштування віддаленого керування. Далі через консоль «Керування комп’ютером» у розділі «Локальні користувачі та групи» до групи «Адміністратори» додається об’єкт Центрального сервера. Наступним кроком у командному рядку виконується команда wecutil qc з підтвердженням дії. Після запуску утиліти eventvwr.exe створюється нова підписка у розділі «Підписки», де вказуються її ім’я та опис. Цільовим журналом визначаються «Переслані події». Як показано на рисунку 11.45, через меню «Вибрати комп’ютер» додаються віддалені сервери для збору подій. Процес завершується визначенням критеріїв фільтрації через опцію «Вибрати події» та налаштуванням облікового запису комп’ютера у меню «Додатково» [13].

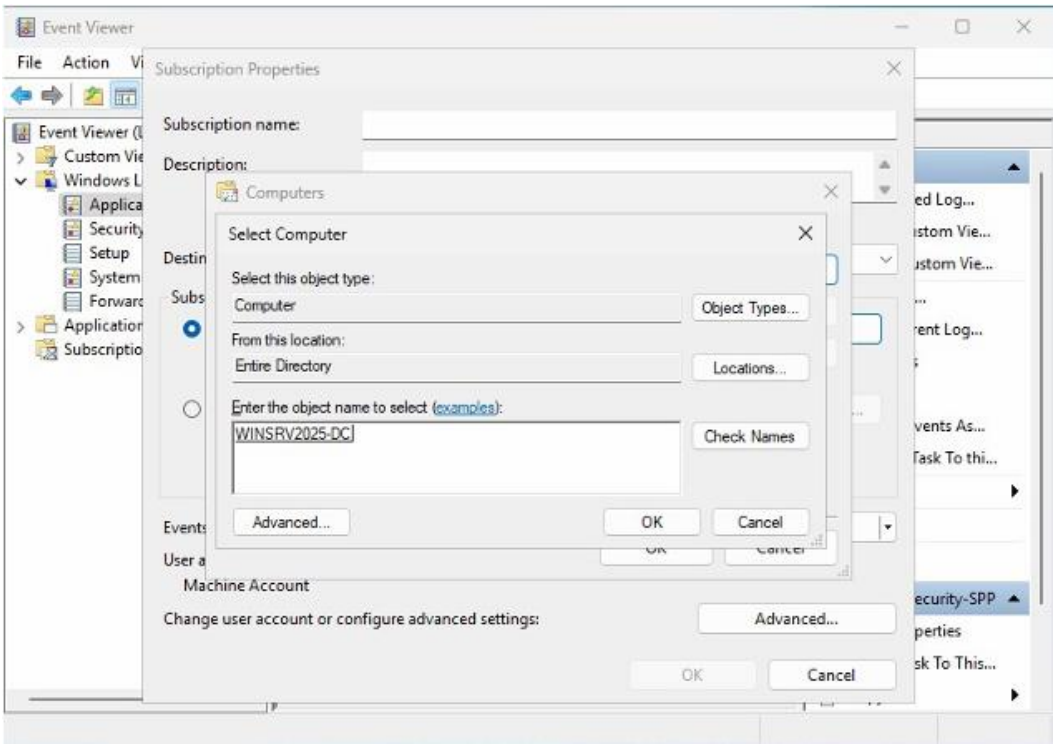

Рисунок 11.45 – Додавання віддаленого сервера для збору подій [13]

Для уточнення масиву даних у журналах використовується механізм фільтрації. Після запуску консолі eventvwr.msc здійснюється перехід до розділу «Журнали Windows», де обирається необхідна категорія (наприклад, Система або Безпека). На панелі дій активується функція «Фільтрувати поточний журнал», інтерфейс якої представлено на рисунку 11.46. У вікні налаштувань визначаються специфічні параметри, такі як рівень події, джерело або ключові слова, що дозволяє адаптувати відображення записів до поточних діагностичних потреб. Застосування фільтра відбувається після підтвердження налаштувань, що призводить до відображення уточненого списку подій.

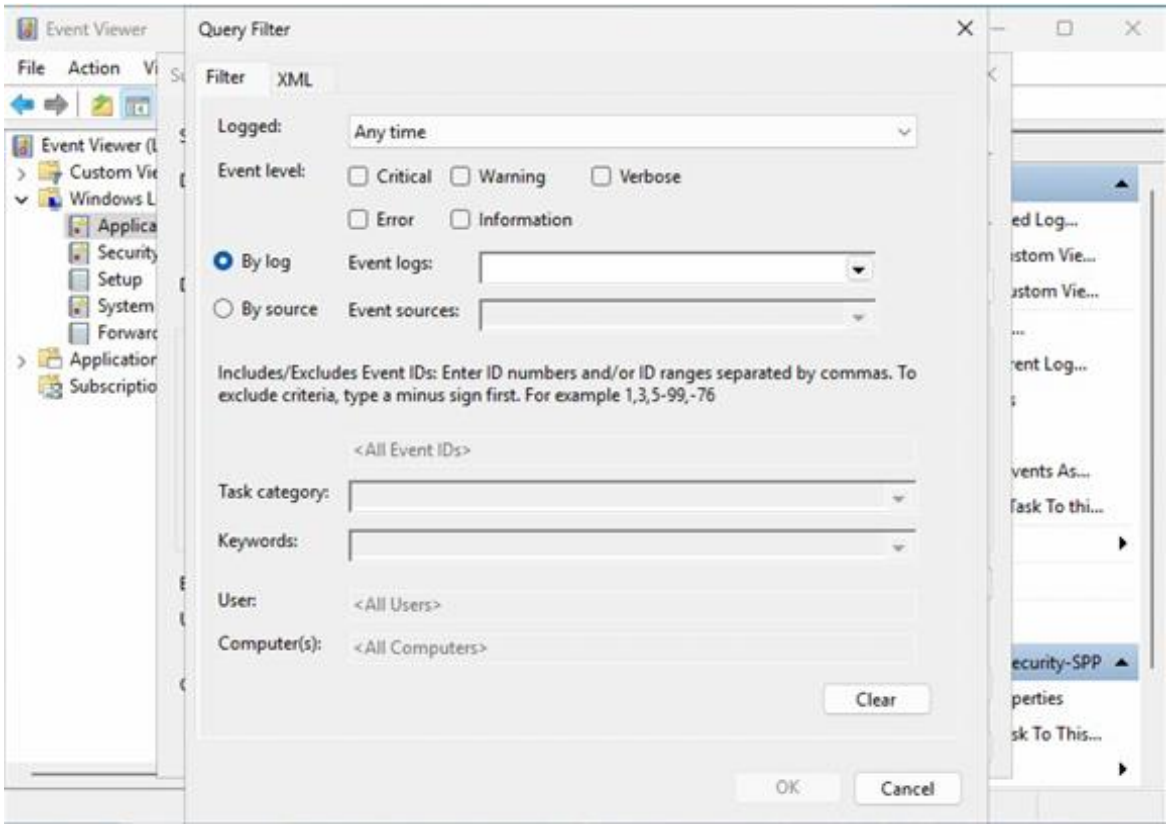

Рисунок 11.46 – Фільтрація журналів засобу перегляду подій [13]

Зміна стандартного розташування файлів журналів у Windows Server 2025 виконується шляхом редагування системного реєстру. Процедура ініціюється запуском редактора реєстру командою regedit. Для зміни шляху системного журналу необхідно перейти до гілки HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\EventLog\System. У даному розділі модифікується значення параметра File, де вказується новий шлях до файлу, як це продемонстровано на рисунку 11.47. Аналогічні дії виконуються у відповідних гілках для журналів програм (...EventLog\Application) та журналів безпеки (...EventLog\Security). Коректне виконання цих маніпуляцій дозволяє перенаправити потік запису подій на інші носії, що є важливою складовою стратегії керування дисковим простором сервера.

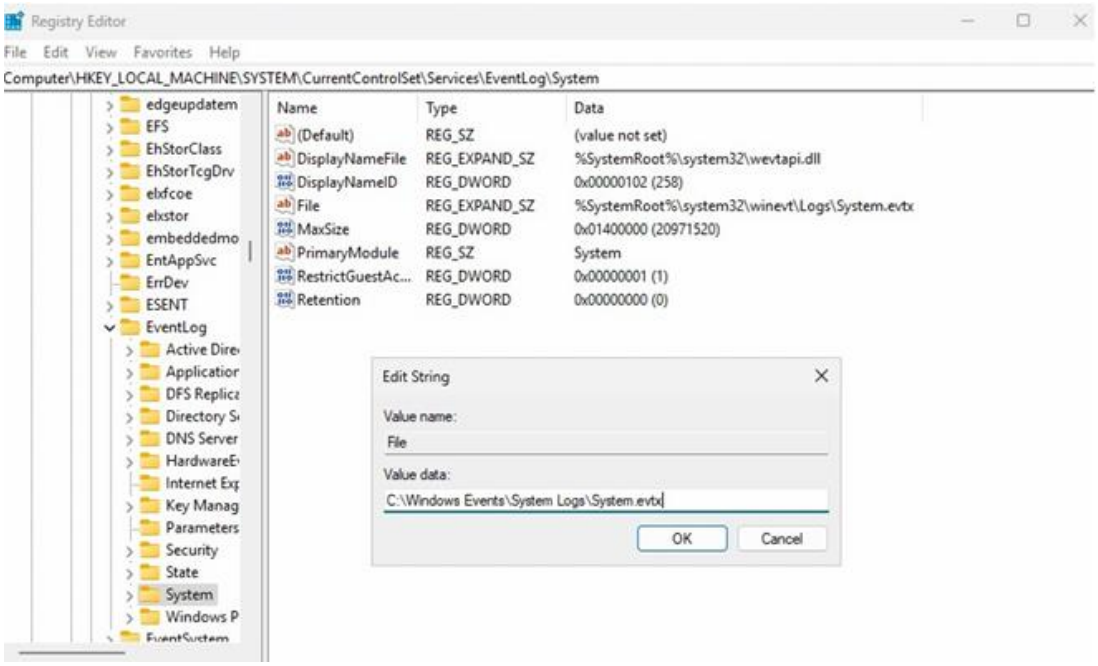

Рисунок 11.47 – Зміна розташування журналів за замовчуванням [13]

Тема 12
Active Directory

Вступ до служб доменів Active Directory

Технології Active Directory від корпорації Microsoft пройшли значний шлях розвитку з моменту їх первинного випуску у складі Windows Server 2000. Трансформувавшись із єдиного продукту, який позначався просто як Active Directory або AD, у середовищі Windows Server 2025 даний напрямок тепер охоплює загалом п'ять окремих технологій Active Directory. Кожна з цих технологій має подібну природу – всі вони існують для забезпечення функціонування служб каталогів та слугують платформою для майбутньої інтеграції технологій Microsoft. Чотирма додатковими ролями служб Active Directory у Windows Server 2025 є: полегшені служби каталогів Active Directory (Active Directory Lightweight Directory Services – AD LDS), служби федерації Active Directory (Active Directory Federation Services – AD FS), служби сертифікації Active Directory (Active Directory Certificate Services – AD CS) та служби управління правами Active Directory (Active Directory Rights Management Services – AD RMS) [13].

Проте, завжди основна увага зосереджується на традиційній службі Active Directory – доменних службах Active Directory (Active Directory Domain Services – AD DS). Розглядається інформація, необхідна для розуміння сутності AD DS та причин, через які ця технологія стала найбільш поширеною платформою корпоративних каталогів, що використовується на сьогоднішній день.

Проектування Active Directory

Процес проектування Active Directory визначається як етап, що передує будь-якому фізичному впровадженню серверного обладнання. Архітектура каталогу повинна розроблятися з урахуванням довгострокової перспективи, оскільки зміни в базовій структурі після розгортання є ресурсомісткими та технічно складними. Проектування традиційно розділяється на два взаємопов'язані рівні: логічне проектування, що визначає адміністративні межі та ієрархію безпеки, та фізичне проектування, яке відображає топологію мережі та розміщення контролерів домену. Головною метою проектування визначається створення масштабованої системи, що задовольняє вимоги безпеки та бізнес-потреби організації.

На етапі проектування логічної структури першочерговим завданням є визначення кількості лісів та доменів. Ліс розглядається як найвища межа безпеки, і згідно з сучасними рекомендаціями, модель єдиного лісу (Single Forest Model) визнається оптимальною для більшості підприємств. Такий підхід забезпечує спрощене управління, єдину схему та глобальний каталог, мінімізуючи адміністративні витрати на синхронізацію. Створення кількох лісів проектується лише за наявності специфічних вимог до ізоляції, які неможливо вирішити засобами одного лісу. Стосовно структури доменів, тенденція проектування змістилася в бік спрощення: модель єдиного домену вважається найбільш доцільною, оскільки сучасні можливості AD DS дозволяють ефективно керувати мільйонами об'єктів без необхідності створення додаткових доменів виключно з технічних причин [20-23].

Важливим аспектом проектування є інтеграція з системою доменних імен (DNS), яка слугує основою для функціонування Active Directory. При виборі простору імен необхідно визначити стратегію, що запобігає конфліктам між внутрішніми та зовнішніми іменами. У сучасних реалізаціях рекомендується використання виділеного субдомену публічного простору (наприклад, corp.company.com) для внутрішньої інфраструктури AD. Використання немаршрутизованих суфіксів, таких як .local, більше не вважається кращою практикою через потенційні проблеми із сумісністю сертифікатів безпеки та інтеграцією з хмарними сервісами. Правильно спроектований простір імен забезпечує коректну реєстрацію службових записів та безперебійну роботу механізмів локалізації служб.

Проектування структури організаційних підрозділів (OU) виконується з метою забезпечення делегування адміністративних повноважень та ефективного застосування групових політик (GPO). Ієрархія OU не обов'язково повинна віддзеркалювати організаційну структуру підприємства; натомість вона проектується на основі адміністративної моделі. Розробляються схеми, що дозволяють застосовувати політики безпеки до конкретних типів

об’єктів (користувачів, комп’ютерів, серверів) або географічних одиниць. Гнучкість структури OU дозволяє уникнути створення нових доменів для розмежування прав доступу, що значно спрощує загальну архітектуру [20-23].

Фізичне проектування зосереджується на оптимізації мережевого трафіку через конфігурацію сайтів та зв’язків між ними. Сайт в Active Directory визначається як сукупність підмереж з високошвидкісним з’єднанням. Правильне проектування топології сайтів є необхідним для контролю трафіку реплікації, який між сайтами піддається стисненню та плануванню, а також для локалізації трафіку автентифікації, щоб клієнтські комп'ютери зверталися до найближчих контролерів домену. Цей етап проектування є сполучною ланкою між логічними компонентами каталогу та реальною мережевою інфраструктурою.

Розуміння інфраструктури Active Directory у Windows Server 2025

Active Directory (AD) визначається як фундаментальна технологія від корпорації Microsoft, що виконує функцію розподіленої служби каталогів. Вона є необхідною для організації та управління мережевими ресурсами в ієрархічний та безпечний спосіб. Система діє як централізоване сховище, де зберігаються критично важливі об’єкти, такі як облікові записи користувачів, комп’ютери, принтери та мережеві служби, кожен з яких має власні унікальні налаштування безпеки. Унікальні атрибути кожного об’єкта в межах AD уможливлюють гранулярний контроль над управлінням ресурсами, що дозволяє здійснювати точне адміністрування всією мережею. Наприклад, кожен об’єкт, будь то обліковий запис користувача, комп’ютер, принтер або мережева служба, володіє специфічними атрибутами, включаючи ідентифікатори безпеки (Security Identifiers – SIDs), членство в групах та списки контролю доступу (Access Control Lists – ACL). Наявність цих атрибутів надає адміністраторам повноваження визначати індивідуальні дозволи, ролі та політики доступу, гарантуючи, що заходи безпеки та функціональні можливості адаптовані до вимог кожного об’єкта.

Архітектура AD, як проілюстровано на рисунку 12.1, структурується навколо трьох фундаментальних рівнів: домену, який є базовою одиницею адміністрування, що забезпечує межу для політик та налаштувань безпеки; дерева, що являє собою колекцію доменів, пов’язаних безперервним простором імен, відображаючи ієрархічні відносини між ними; та лісу – найвищого рівня організації, який може охоплювати кілька дерев і слугує верхнім шаром, що інтегрує всю службу каталогів [13].

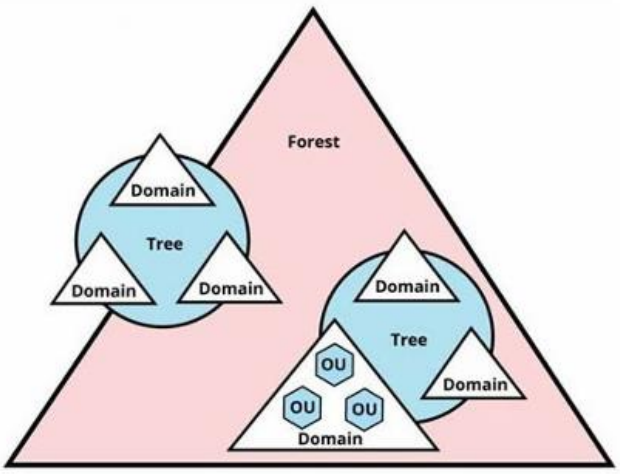

Рисунок 12.1 – Архітектура AD [13]

Такий багаторівневий підхід сприяє ефективному управлінню ресурсами та масштабованості, одночасно підтримуючи складні організаційні структури. Це дозволяє підприємствам налаштовувати свою мережеву інфраструктуру відповідно до специфічних операційних потреб, зберігаючи при цьому надійну безпеку та адміністративний нагляд. У подальшому викладі досліджуються специфічні функції та конфігурації AD, що забезпечує отримання знань та навичок, необхідних для ефективного впровадження та управління службами каталогів в організації.

Active Directory розглядається не просто як служба каталогів, а як основа сучасних IT-інфраструктур, що відіграє критичну роль в управлінні середовищами Windows. Для фахівців-початківців розуміння значущості AD є необхідним для усвідомлення її функціональних можливостей та переваг.

По-перше, однією з головних переваг визначається здатність централізувати управління. IT-адміністратори можуть керувати користувачами, комп'ютерами та ресурсами з єдиного місця, що значно знижує складність та адміністративні витрати. Цей централізований підхід оптимізує процес надання та скасування доступу користувачів, полегшуючи підтримку організованої та ефективної мережі.

По-друге, AD підвищує рівень безпеки завдяки використанню SIDs та ACLs. Забезпечуючи доступ до конкретних ресурсів лише авторизованим користувачам, AD захищає конфіденційну інформацію та знижує ризик несанкціонованого доступу. Ця модель безпеки є критично важливою для захисту організаційних даних та дотримання нормативних стандартів.

По-третє, ієрархічна структура AD підтримує масштабованість організацій. У міру зростання бізнесу AD дозволяє безперешкодно інтегрувати нових користувачів та ресурси без шкоди для продуктивності або безпеки, що надає можливість адаптуватися до змінних потреб та ефективно розширювати IT-інфраструктуру.

Нарешті, AD сприяє застосуванню групових політик у всій мережі, що дозволяє організаціям уніфіковано впроваджувати налаштування безпеки та стандарти відповідності. Ця можливість гарантує дотримання політик організації всіма користувачами та пристроями, підвищуючи загальний стан безпеки та операційну ефективність. Розуміння цих основних принципів дозволяє оцінити ключову роль AD в управлінні та захисті мережевих ресурсів, що закладає фундамент для ефективного IT-адміністрування.

Функціонування Active Directory спирається на декілька критично важливих протоколів та служб, кожен з яких робить внесок у різні аспекти управління мережею та безпеки. Lightweight Directory Access Protocol (LDAP) є протоколом, який відіграє вирішальну роль у забезпеченні можливості запитів та взаємодії користувачів і додатків з даними каталогу. LDAP надає стандартизований метод доступу та управління інформацією, що зберігається в AD, що робить його ключовим елементом операцій служби каталогів [13].

Іншим важливим компонентом є Kerberos – складний механізм автентифікації, що лежить в основі інфраструктури безпеки AD. Kerberos використовує систему квитків для безпечної перевірки особистості користувачів і серверів у мережі, запобігаючи несанкціонованому доступу та гарантуючи належну автентифікацію всіх суб'єктів. Протокол DNS також є невід'ємною частиною функціональності AD. Він слугує каталогом для Інтернету та внутрішніх мереж, перетворюючи зручні для користувача доменні імена на цифрові IP-адреси. Цей процес трансляції є необхідним для ефективного пошуку та доступу до мережевих ресурсів. У середовищі AD DNS не лише перетворює доменні імена, але й підтримує специфічні функції, такі як визначення розташування контролерів домену та забезпечення доступності служб у мережі. Разом ці протоколи та служби формують основу AD, дозволяючи надавати безпечну, масштабовану та ефективну службу каталогів [13].

Active Directory є потужною платформою, що надає комплексні послуги для забезпечення централізованого управління мережевими ресурсами, оптимізуючи роботу системних адміністраторів. Для ефективного управління різними аспектами служб AD корпорацією Microsoft пропонується набір адміністративних консолей у рамках Microsoft Management Console (MMC) (mmc.exe), кожна з яких адаптована до конкретних завдань. Центр адміністрування Active Directory (Active Directory Administrative Center – dsac.exe), зображений на рисунку 12.2, є ключовим інструментом в управлінні службами каталогів Windows Server.

Рисунок 12.2 – Центр адміністрування Active Directory у Windows Server 2025 [13]

Цей сучасний інтерфейс інтегрує кілька функцій управління, дозволяючи адміністраторам ефективно контролювати сервіси. Він включає оснастку «Active Directory – користувачі й комп'ютери» (dsa.msc), яка є необхідною для управління обліковими записами користувачів, об'єктами комп'ютерів, організаційними підрозділами та пов'язаними з ними властивостями. Цей інструмент є фундаментальним для повсякденних адміністративних завдань, таких як створення та управління користувачами, групами та пристроями, а також їх організація в структурі AD.

Консоль Active Directory Domains and Trusts (domain.msc) використовується для завдань, пов'язаних з управлінням доменами. Вона дозволяє налаштовувати та керувати довірою доменів, що є критично важливим для забезпечення безпечних комунікацій та спільного використання ресурсів між різними доменами, а також керує функціональними рівнями домену. Консоль Active Directory Sites and Services (dssite.msc) є критично важливим інструментом для управління реплікацією між різними сайтами AD, що представляють фізичну структуру мережі. Інструмент дозволяє оптимізувати та контролювати реплікацію інформації каталогу між різними географічними локаціями. Окрім графічних інструментів, модуль AD для Windows PowerShell пропонує інтерфейс командного рядка для більш просунутих та автоматизованих завдань управління. Командлети PowerShell дозволяють створювати сценарії для складних операцій, автоматизувати повторювані завдання та керувати об'єктами AD у великих масштабах [20-23].

Для розгортання служб каталогів в організації необхідно встановити та налаштувати роль AD DS на сервері Windows. AD DS є основою середовища AD, забезпечуючи зберігання, організацію та управління інформацією про мережеві ресурси. Вона також підтримує розширені функції безпеки, такі як централізована автентифікація та авторизація. Варто зазначити, що доступ до великої кількості безкоштовних сценаріїв PowerShell можна отримати в Microsoft Script Center та PowerShell Gallery. Ці платформи слугують відомими репозиторіями, де IT-фахівці можуть знаходити та ділитися сценаріями для різних адміністративних завдань, включаючи ті, що стосуються AD та DNS [21].

Додавання та налаштування ролі AD DS (Active Directory Domain Services)

У середовищах Windows Server роль AD DS визначається як критично важлива для надання централізованих служб каталогів, що полегшують управління мережею та автентифікацію. Ппроцес додавання та налаштування ролі AD DS охоплює ключові завдання, такі як розгортання контролерів домену (DC), налаштування та управління доменами, а також створення ієрархічних структур, таких як дерева та дочірні домени (рис. 12.3).

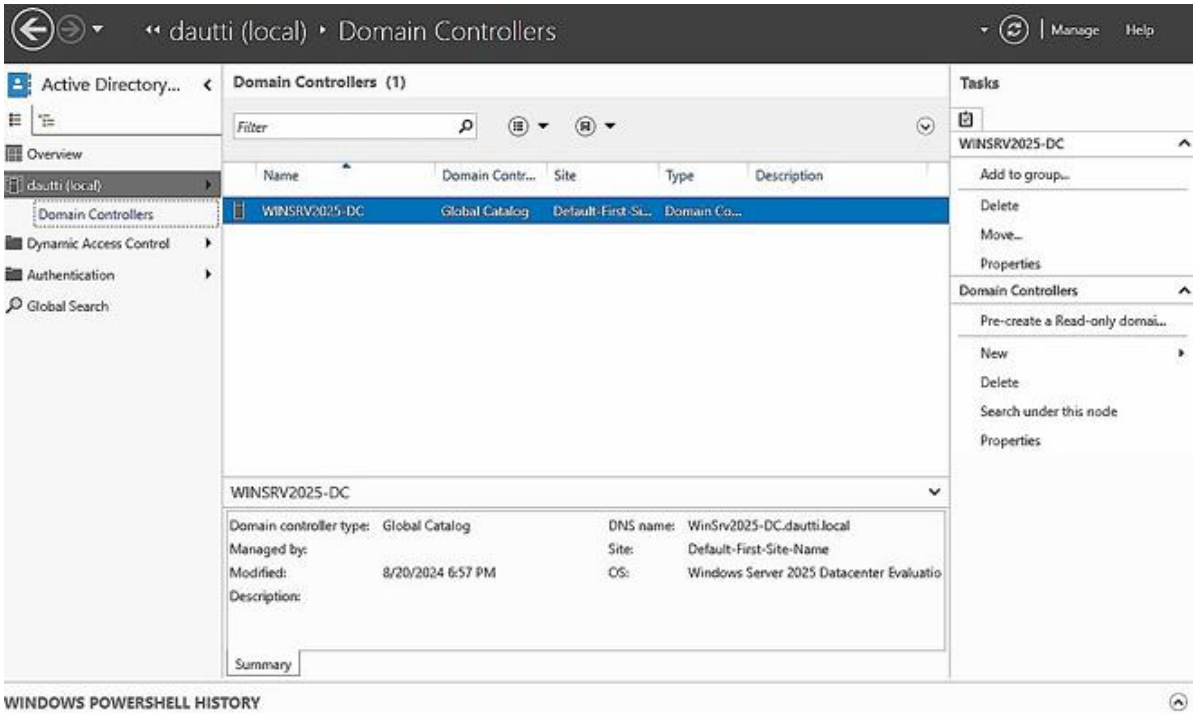

Рисунок 12.3 – Доступ до контролерів домену через Центр адміністрування Active Directory [13]

Контролер домену (DC), як зображено на рисунку 12.3, є сервером, що відіграє критичну роль в управлінні та перевірці ідентичності користувачів у мережі організації. Його основною функцією є автентифікація користувачів та авторизація доступу до мережевих ресурсів на основі політик безпеки, визначених у домені [20].

У ранніх середовищах Windows, зокрема Windows NT, управління доменом покладалося на первинний контролер домену (PDC) для виконання основних функцій та резервні контролери домену (BDC) для забезпечення надлишковості. Проте цю модель було замінено моделлю реплікації з багатьма майстрами (multi-master replication), впровадженою у Windows 2000, що дозволило кільком DC розподіляти відповідальність за управління функціями домену. Такий підхід підвищує надійність та доступність, оскільки всі DC можуть виконувати операції читання та запису, гарантуючи стійкість та доступність служб автентифікації та каталогів у всій мережі. Windows Server 2025 революціонізував підхід до DC, усунувши традиційні ролі первинного та резервного серверів. Натомість DC тепер ідентифікуються за послідовними номерами, наприклад DC1 та DC2, що вказує на їхню черговість, а не функцію. Цей сучасний підхід створює умови для більш гнучкого та масштабованого середовища управління доменом, де всі DC вважаються рівноправними партнерами, що поділяють відповідальність за автентифікацію та служби каталогів. Варто зазначити, що коли сервер приєднується до домену, але не бере на себе роль DC, він класифікується як рядовий сервер. Такі сервери працюють відповідно до політик домену та контролю доступу, але не обробляють запити на автентифікацію та не виконують завдань з управління доменом. Враховуючи, що DC є центральними елементами доступу до домену та автентифікації, розуміння концепції доменів є необхідним для усвідомлення повного обсягу інфраструктури AD [13].

Домени визначаються як фундаментальні компоненти управління мережею, що організовують та групують користувачів, комп’ютери, пристрої та мережеві служби в рамках єдиної адміністративної структури. Це логічне групування дозволяє здійснювати централізоване управління ресурсами та політиками безпеки. DC є критично важливим у цій конфігурації, а AD DS відіграє ключову роль у встановленні та підтримці функціональності домену. На рисунку 12.4 продемонстровано процес налаштування домену у вікні майстра конфігурації Active Directory Domain Services, що ілюструє створення та управління доменами.

Критично важливо розрізняти домен каталогу та доменне ім’я. У контексті служб каталогів домен стосується структурованої бази даних мережевих ресурсів, включаючи

користувачів, сервери та пристрої, які керуються колективно відповідно до специфічних адміністративних політик. Цей домен сприяє ефективному управлінню та безпеці в IT-інфраструктурі організації. З іншого боку, доменне ім'я є частиною DNS – ієрархічної системи імен, що використовується для ідентифікації та локалізації ресурсів в Інтернеті, таких як веб-сайти та сервери електронної пошти. Крім того, домени можуть бути організовані в дерево доменів, яке представляє ієрархічну структуру з кількох доменів. Ця структура дозволяє організувати домени у відносини «батько-дитина», де кожен домен у дереві може успадковувати політики та налаштування від свого батьківського домену, зберігаючи при цьому власну конфігурацію [13].

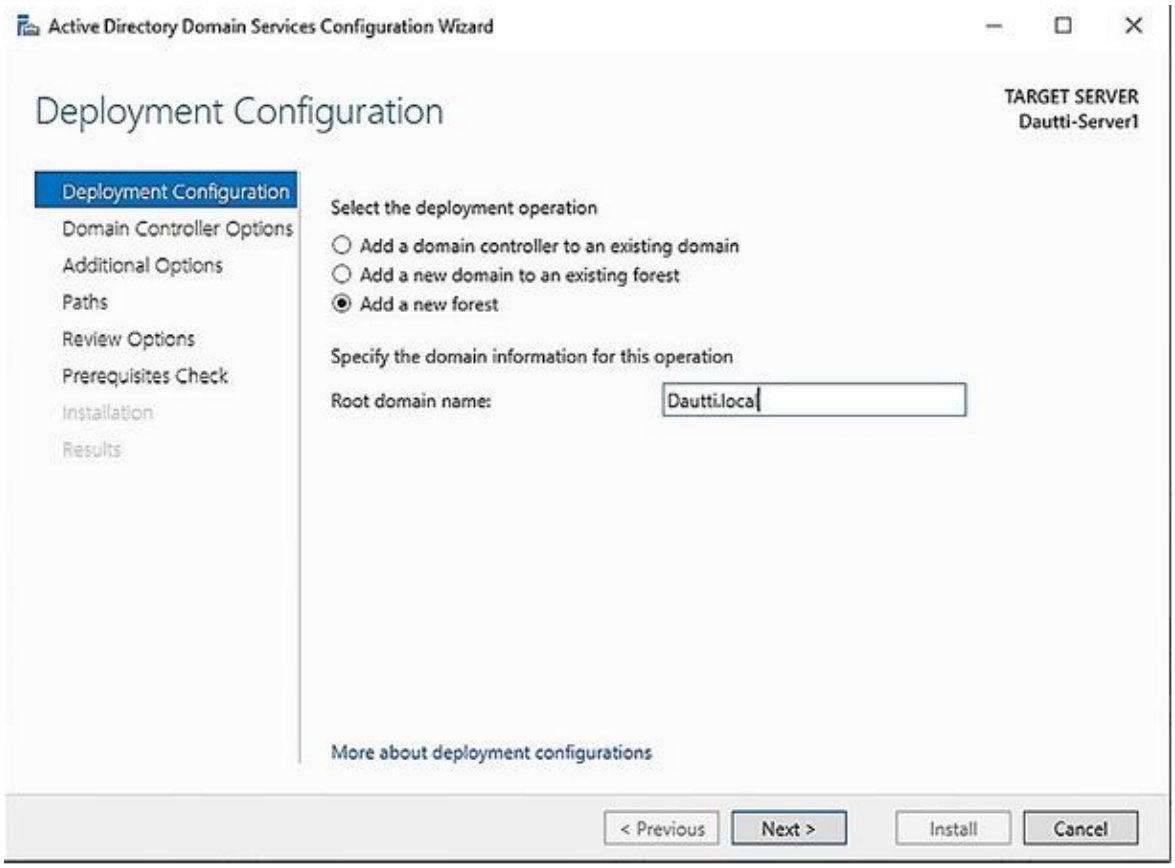

Рисунок 12.4 – Налаштування домену у вікні майстра конфігурації Active Directory Domain Services (налаштування кореневого домену) [13]

Для повного розуміння архітектури AD необхідно розглянути концепцію дерева доменів. Дерево доменів представляє логічну структуру в межах AD, що складається з одного або кількох доменів, які мають спільний простір імен і розташовані ієрархічно. Таке ієрархічне налаштування не лише організовує домени, але й забезпечує їхню взаємну довіру завдяки транзитивним довірчим відносинам. В AD довірчі відносини дозволяють користувачам в одному домені проходити автентифікацію та отримувати доступ до ресурсів в іншому домені без необхідності використання окремих облікових даних.

Транзитивна довіра означає, що якщо домен A довіряє домену B, а домен B довіряє домену C, то домен A автоматично довірятиме домену C. Ця вбудована довіра спрощує спільне використання ресурсів та автентифікацію між доменами в межах одного дерева. Процес впровадження нового домену в існуюче дерево передбачає вказання імені батьківського домену під час підвищення сервера для встановлення відповідної ієрархії. Це додавання дозволяє новому домену успадковувати політики та налаштування від батьківського, встановлюючи власну унікальну ідентичність у дереві. Рисунок 12.5 надає візуальне представлення створення деревоподібного домену у Windows Server 2025 [13].

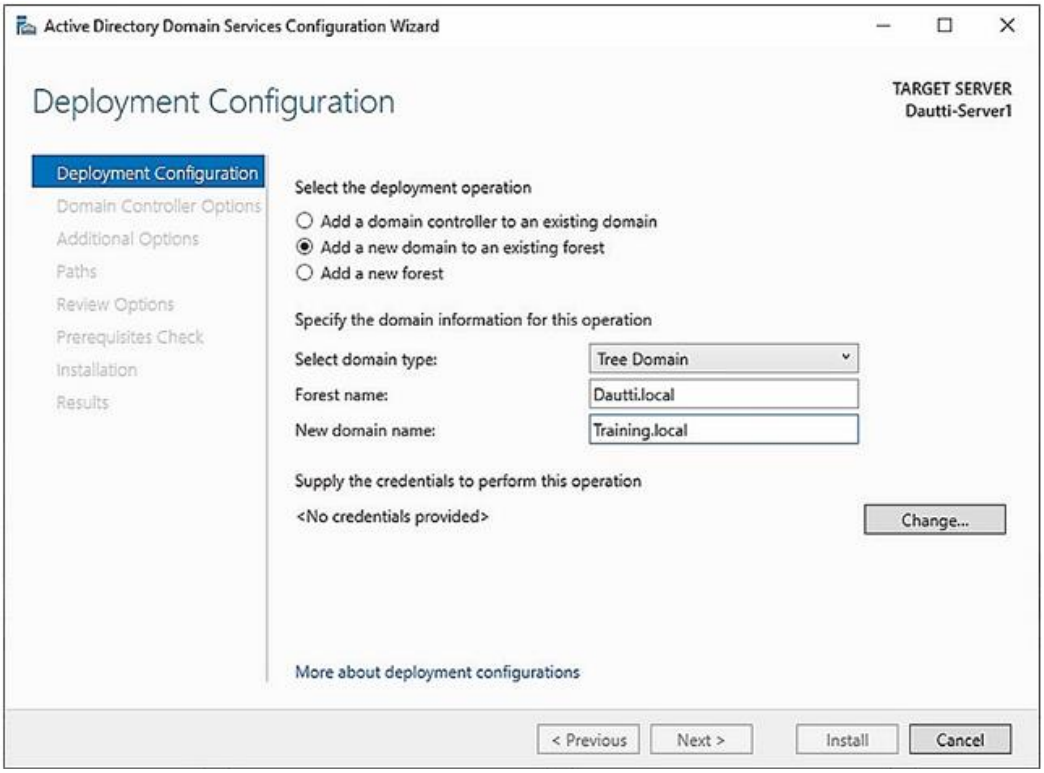

Рисунок 12.5 – Налаштування деревоподібного домену в Windows Server 2025 [13]

Концепція дерева доменів розширюється, коли кілька дерев доменів об'єднуються для формування лісу. Ліс представляє ширшу організаційну структуру, що групує всі дерева доменів підприємства, дозволяючи створити єдине середовище каталогів (рис. 12.6).

В AD концепція лісу аналогічна природному лісу, який складається з багатьох дерев. Ліс AD може складатися з одного дерева доменів або колекції взаємопов'язаних дерев. Кожне дерево доменів у лісі має спільну схему та глобальний каталог, але самі дерева не обов'язково повинні мати спільний простір імен.

Кореневий домен – це перший домен, створений у дереві доменів, що слугує основою для всієї структури. Він часто виконує критичні ролі, такі як майстер схеми та майстер іменування доменів. Дерево доменів, що діє як кореневий домен, може існувати незалежно в лісі, але за наявності кількох дерев ліс діє як всеосяжна структура, що інтегрує та керує цими деревами, створюючи цілісне та масштабоване середовище каталогів [13].

Ліс виступає як найвищий рівень структури AD, забезпечуючи єдину систему каталогів. Для створення та налаштування лісу у Windows Server 2025 використовується майстер конфігурації AD DS, який також застосовується для налаштування дерев доменів.

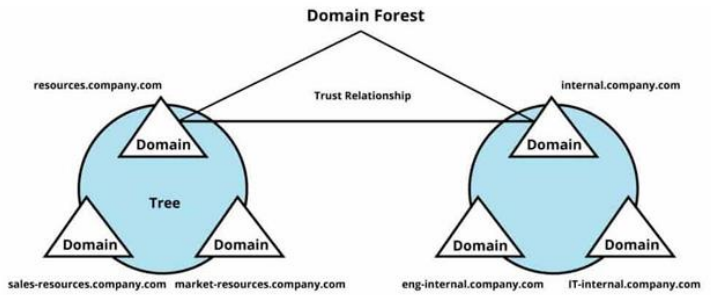

Рисунок 12.6 – Ієрархічна архітектура лісу доменів [13]

У рамках структури дерева доменів можуть бути створені додаткові субдомени, що називаються дочірніми доменами. Вони функціонують як підрозділи батьківського дерева доменів, дозволяючи більш гранулярну організацію та управління ресурсами. Дочірній домен є підпорядкованим доменом у структурі дерева AD. Наприклад, якщо Dautti.local слугує кореневим доменом лісу, встановлюючи базовий простір імен, то Administration.Dautti.local

може бути створений як дочірній домен. Він є розширенням простору імен батьківського домену, забезпечуючи цілісну ієрархію каталогів. Створення дочірнього домену у Windows Server 2025 виконується за допомогою майстра конфігурації AD DS, як проілюстровано на рисунку 12.7.

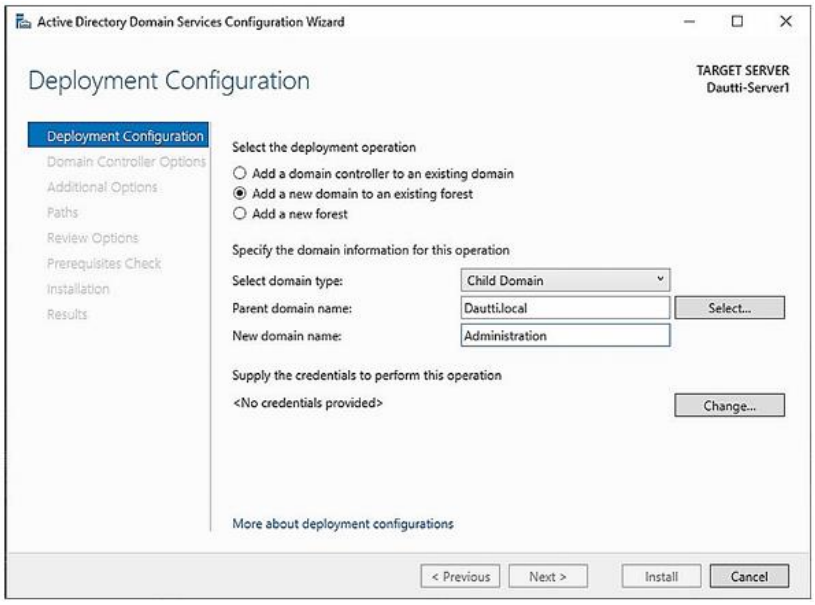

Рисунок 12.7 – Налаштування дочірнього домену в Windows Server 2025 [13]

AD DS є надійною системою, що вимагає ретельного планування. Ключовим компонентом AD DS є ролі майстрів операцій (Operations Master Roles), необхідні для ефективного управління службами каталогів. При встановленні ролі AD DS та підвищенні сервера до DC автоматично призначаються п’ять критичних ролей майстрів операцій, розділених на дві категорії [13].

Ролі рівня лісу (Forest-wide roles) включають майстра схеми (Schema Master), який контролює схему каталогу, та майстра іменування доменів (Domain Naming Master), що керує простором імен та гарантує унікальність імен доменів у лісі.

Ролі рівня домену (Domain-wide roles) включають RID-майстра (RID Master), що виділяє ідентифікатори безпеки (SIDs), емулятора PDC (PDC Emulator), який обробляє зміни паролів та синхронізацію часу, та майстра інфраструктури (Infrastructure Master), відповідального за оновлення посилань на об’єкти в інших доменах. На рисунку 12.8 проілюстровано структуру AD DS, де кореневий домен утримує ролі рівня лісу, а кожен домен має власні ролі рівня домену.

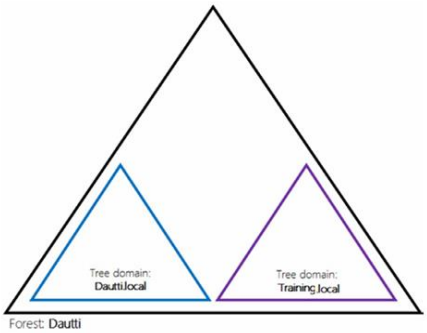

Рисунок 12.8 – Структура AD DS [13]

Ці п’ять ролей відомі як ролі FSMO (Flexible Single Master Operations). Термін «гнучкий» відображає можливість їх перенесення на інші DC, а «єдиний» вказує, що лише один DC може утримувати кожну роль у певний момент часу для запобігання конфліктам.

Для ефективного розрізнення домену та робочої групи (workgroup) важливо розуміти базові архітектури мережі: одноранговy (P2P) та клієнт-серверну. У P2P-мережі, або робочій

групі, кожен комп'ютер працює незалежно та керує власними ресурсами без централізованого контролю. Це підходить для малих мереж, але ускладнює управління зі зростанням кількості пристроїв. Натомість клієнт-серверна мережа, або домен, забезпечує структурований підхід, де центральний сервер (DC) контролює адміністративні завдання та забезпечує виконання політик безпеки. Централізоване управління є критичним для великих організацій, підтримуючи масштабованість та узгодженість політик.

Ключовою концепцією в AD є довірчі відносини, які відіграють важливу роль у взаємодії комп'ютерів, DC та доменів. Коли комп'ютер інтегрується в домен, він переходить від використання локального менеджера облікових записів безпеки (SAM) до системи автентифікації DC, зазвичай Kerberos. Це централізує автентифікацію та підвищує безпеку. Довірчі відносини поширюються і на рівень лісу, де кожен домен автоматично довіряє методам автентифікації інших доменів, створюючи єдину структуру безпеки (як проілюстровано на прикладі взаємодії доменів у лісі) [13].

Функціональні рівні в AD є критичними елементами, що визначають сумісність та поведінку середовища. Існує два основні рівні: функціональний рівень лісу (FFL) та функціональний рівень домену (DFL) (рис. 12.9).

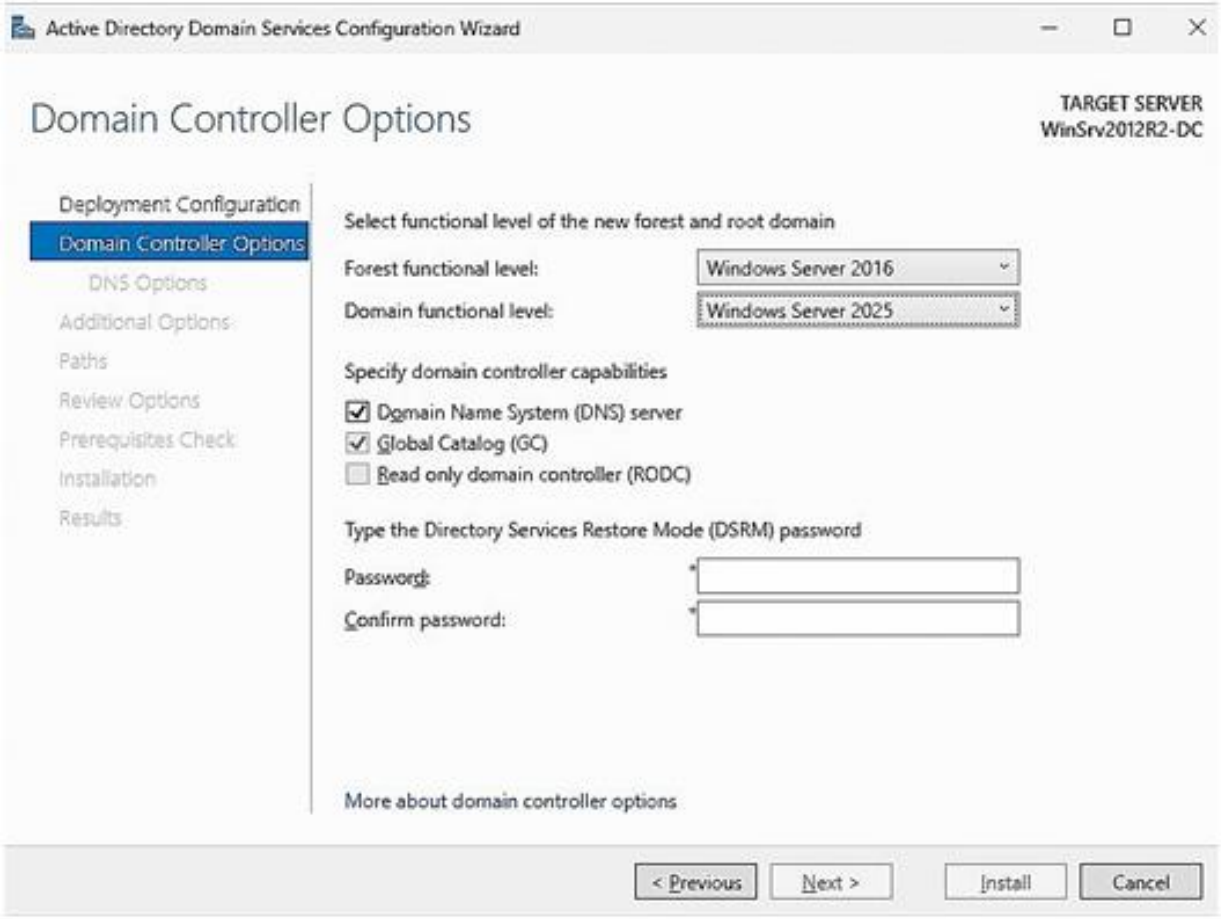

Рисунок 12.9 – FFL та DFL у Windows Server 2025 [13]

FFL визначає, які версії Windows Server можуть працювати на DC у всьому лісі, та розблоковує функції рівня лісу (наприклад, кошик AD). DFL застосовується до окремих доменів, визначаючи підтримувані версії серверів та функції рівня домену. У контексті Windows Server 2025 мінімальні рівні можуть бути встановлені на Windows Server 2016, що забезпечує сумісність. Підвищення рівнів до Windows Server 2025 активує найсучасніші функції. Важливо зазначити, що після підвищення функціонального рівня його не можна знизити. Для перевірки та управління рівнями використовується інструмент «Active Directory – домени і довіра» (Active Directory Domains and Trusts), де у властивостях кореневого домену відображаються поточні значення, як показано на рисунку 12.10.

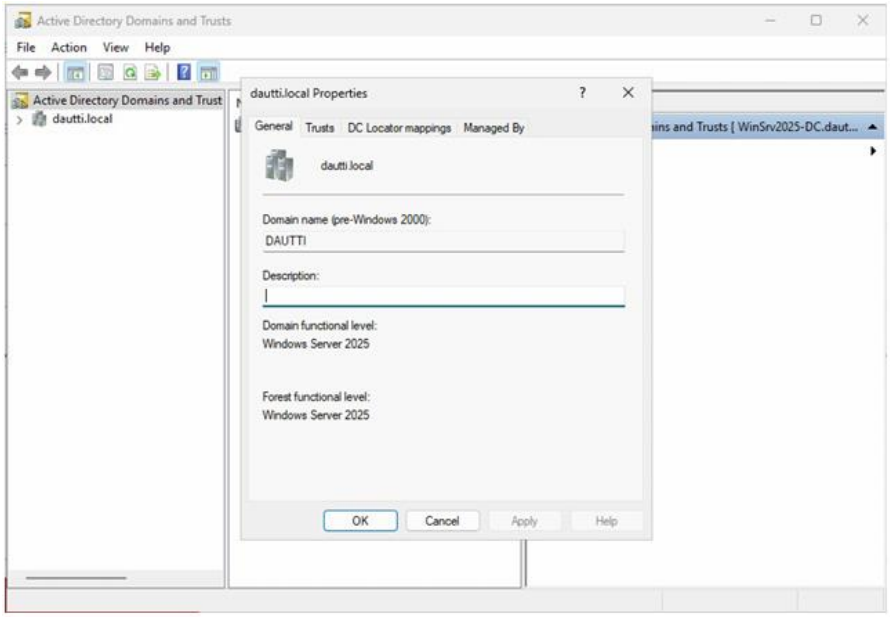

Рисунок 12.10 – Перевірка FFL та DFL [13]

В AD DS концепція простору імен є фундаментальною для організації доменів та лісів. Простір імен слугує логічним ідентифікатором, що унікально називає домен або ліс. Наприклад, як показано на рисунку 12.11, домен Dautti.local функціонує як кореневий домен та ліс, а ITTrainings.local та Administration.local є окремими деревами доменів.

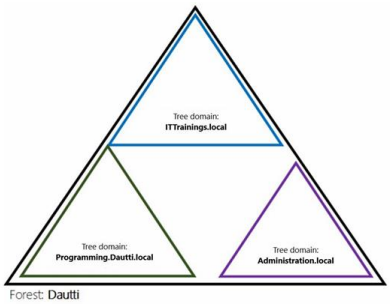

Рисунок 12.11 – Концепція простору імен у AD DS [13]

Спільний компонент Dautti.local вказує на безперервний простір імен, що означає зв’язок усіх доменів через спільну угоду про іменування. Це можна порівняти з системою URL в Інтернеті. Безперервний простір імен забезпечує логічний зв’язок доменів, полегшуючи управління та навігацію [13].

Окрім логічної структури, AD включає фізичну структуру, відому як сайт. Сайт представляє конкретне фізичне розташування в мережевій інфраструктурі та може охоплювати один або кілька доменів, з’єднаних високошвидкісними каналами. Метою визначення сайтів є оптимізація мережевого трафіку, зокрема реплікації та автентифікації. Сайти зменшують непотрібний трафік через глобальні мережі (WAN), обмежуючи реплікацію швидкими локальними каналами. Також AD спрямовує запити автентифікації до DC у тому ж сайті, де знаходиться користувач, що пришвидшує процес входу.

Реплікація в AD є базовою функцією, що забезпечує узгодженість даних на всіх DC лісу. Процес реплікації безперервно поширює зміни (облікові записи, політики), запобігаючи конфліктам. Ефективність реплікації керується топологією, яку автоматично генерує та оптимізує перевірка узгодженості знань (Knowledge Consistency Checker – KCC). KCC створює маршрути реплікації, балансуючи навантаження. Розрізняють внутрішньосайтову (intra-site) реплікацію, що відбувається часто та швидко, та міжсайтову (inter-site) реплікацію, яка є менш частою та оптимізованою для збереження пропускної здатності WAN [13].

На завершення розгляду інфраструктури, схема в AD визначається як критичний компонент, що є проектом організації даних. Вона складається з трьох елементів: об’єктів

(сутності, такі як користувачі), класів (категорії об’єктів, що визначають їх тип) та атрибутів (властивості об’єктів). Схема диктує, як дані зберігаються та організовуються, забезпечуючи стабільність інфраструктури AD.

Керування організаційними одиницями (OU) та типовими контейнерами

Розуміння ролей організаційних одиниць (Organizational Units – OU) та контейнерів визначається, як основа ефективного управління Active Directory. Ці елементи, доступні через консоль «Active Directory – користувачі й комп’ютери» (AD Users and Computers), є невід’ємною частиною організації та адміністрування об’єктів каталогу. OU забезпечують гнучку структуру, дозволяючи адміністраторам створювати ієрархічну організацію в середовищі AD, що полегшує застосування об’єктів групової політики (Group Policy Objects – GPOs) та управління дозволами в різних відділах або групах користувачів. На відміну від них, типові контейнери слугують попередньо визначеними місцями для певних типів об’єктів, таких як користувачі та комп’ютери. Однак їм бракує того ж рівня налаштування та контролю політик, який пропонують OU. У подальшому викладі ці концепції досліджуються глибше, розглядається, яким чином OU можуть бути використані для створення організованої та безпечної інфраструктури AD, а також аналізуються обмеження та використання типових контейнерів. Оволодіння цими компонентами дозволяє адміністраторам підвищити свою здатність ефективно керувати та захищати середовища AD, забезпечуючи добре структурований та зручний для навігації каталог [13].

Організаційні одиниці є критично важливими компонентами в AD, що забезпечують структурований та ефективний підхід до управління користувачами, групами, комп’ютерами та іншими сутностями каталогу. Функціонуючи подібно до папок у файловій системі, OU дозволяють адміністраторам логічно групувати та керувати об’єктами AD на основі організаційних потреб. Це логічне групування є ключовим у спрощенні адміністративних завдань, таких як застосування GPOs та управління дозволами в різних відділах, командах або географічних локаціях організації. Зазвичай структури OU проектуються таким чином, щоб відображати внутрішню бізнес-ієрархію організації, що дозволяє застосовувати адаптований підхід до управління, який узгоджується з її операційною структурою. Кожен домен у лісі AD може встановлювати власну унікальну конфігурацію OU, створюючи гнучку та масштабовану систему, що адаптується до змінних потреб бізнесу. Ця гнучкість є особливо цінною у складних середовищах, де різні домени можуть вимагати відмінних політик та практик управління, як показано на рисунку 12.12.

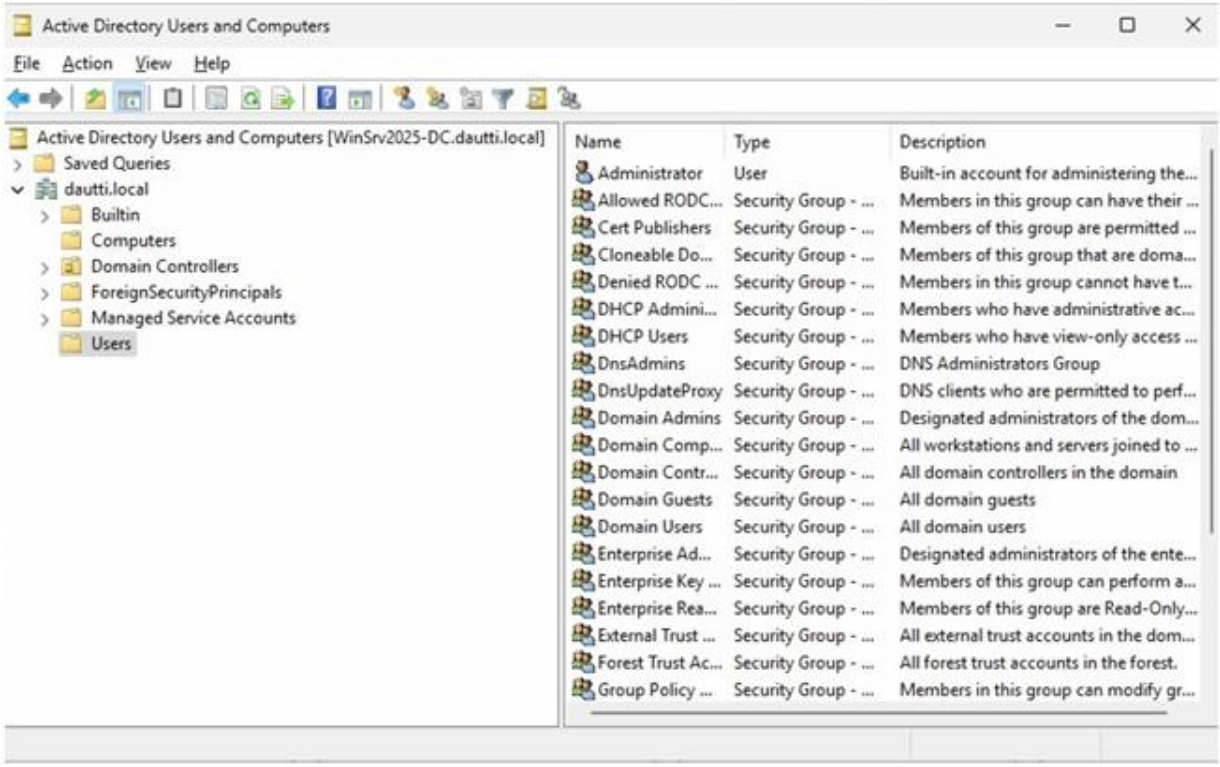

Рисунок 12.12 – Приклад ієрархії організаційних підрозділів у Windows Server 2025 [13]

Окрім OU, необхідно розуміти роль типових контейнерів в AD. Ці контейнери є попередньо визначеними місцями, куди автоматично поміщаються користувачі, комп'ютери та інші об'єкти під час їх створення. Глибоке розуміння попередньо визначених контейнерів є необхідним, коли сервер підвищується до контролера домену (DC). Це підвищення автоматично ініціює створення кількох типових контейнерів, які візуально представлені на рисунку 12.13.

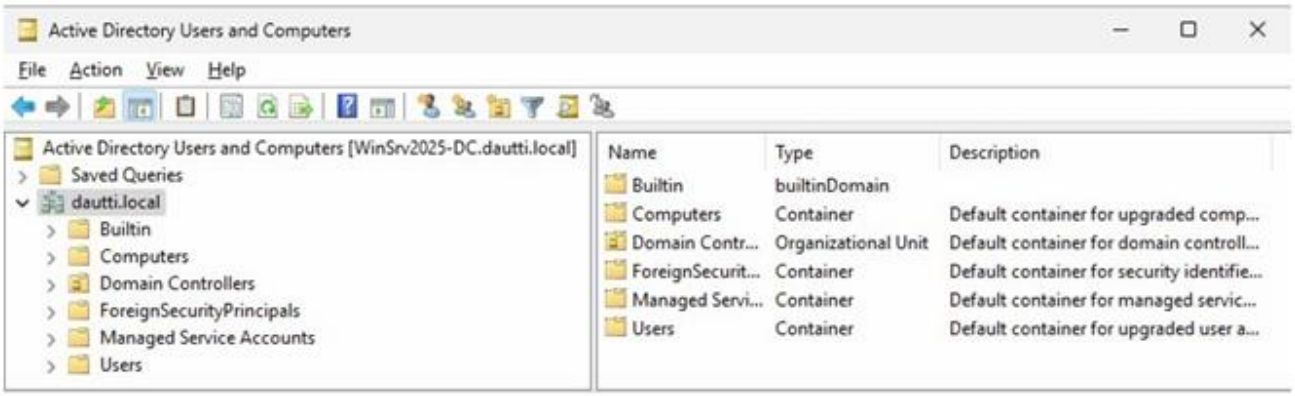

Рисунок 12.13 – Типові контейнери у Windows Server 2025 [13]

Ці контейнери відіграють критичну роль в AD і відрізняються своєю незмінною природою – їх не можна перейменувати, видалити або створити заново, і вони не підлягають зв'язуванню з жодним GPO. Ця незмінність, за задумом, гарантує, що фундаментальні елементи AD залишаються узгодженими та безпечними, тим самим зберігаючи структурну цілісність каталогу. Типові контейнери слугують специфічним цілям, таким як організація користувачів, комп'ютерів та інших об'єктів каталогу стандартизованим способом. Вони забезпечують стабільне середовище для основних операцій AD, гарантуючи, що певні критичні об'єкти завжди зберігаються в передбачуваному місці. Хоча вони не є такими гнучкими, як OU, які можуть бути налаштовані відповідно до потреб організації, типові контейнери залишаються життєво важливими для підтримки фундаментальної структури AD.

Розуміння концепції прихованих типових контейнерів є важливим для системних адміністраторів, навіть якщо ці контейнери не є безпосередньо актуальними для повсякденних завдань. Приховані контейнери виконують значну функцію у підтримці оптимізованого та організованого вигляду в консолі «Active Directory – користувачі й комп'ютери», запобігаючи непотрібному захаращенню, яке могло б ускладнити управління об'єктами AD. Залишаючи певні контейнери поза полем зору, AD забезпечує зручність та керованість інтерфейсу, особливо у великих та складних середовищах. Міркування безпеки також обумовлюють приховування цих контейнерів. Приховані контейнери захищають чутливі системні об'єкти, гарантуючи, що доступ до них мають лише користувачі з відповідними дозволами та знаннями. Цей рівень безпеки допомагає захистити цілісність каталогу та знижує ризик випадкових модифікацій або несанкціонованого доступу до критичних компонентів системи. Для відображення цих прихованих типових контейнерів адміністраторам потрібно увімкнути опцію «Розширені можливості» (Advanced Features) у меню «Вигляд» (View), як зображено на рисунку 12.14 [13].

Активація цієї функції відкриває приховані контейнери, дозволяючи отримати більш повний огляд та розширений контроль над ресурсами каталогу. Ця можливість є особливо цінною для виконання просунутих адміністративних завдань, таких як детальний аудит, тонке налаштування параметрів безпеки або управління об'єктами, які зазвичай не відображаються у стандартному вигляді.

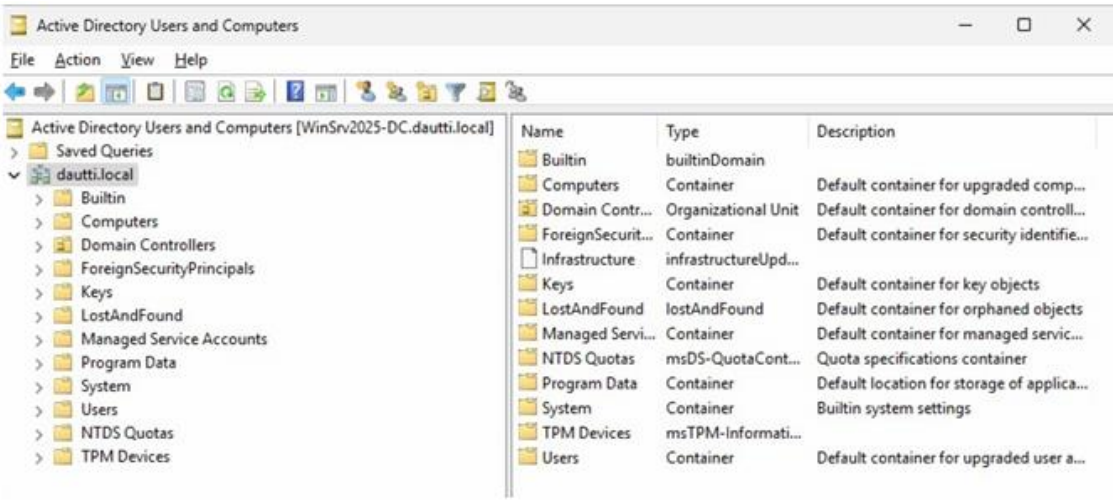

Рисунок 12.14 – Приховані типові контейнери у Windows Server 2025 [13]

Здобувши розуміння цих прихованих типових контейнерів, необхідно розглянути їх практичне застосування та ролі в середовищі AD. У цих контейнерах часто міститься важлива системна інформація, така як об’єкти інфраструктури, суб’єкти безпеки та дані реплікації, які є життєво важливими для безперебійної роботи AD. Розуміння того, як отримати доступ та керувати цими прихованими контейнерами, дозволяє адміністраторам бути повністю підготовленими до підтримки безпечної, ефективної та добре організованої інфраструктури каталогу.

Типові контейнери у Windows Server 2025 є невід’ємною частиною організації та управління об’єктами AD, кожен з яких слугує окремій меті. Контейнер Computers є типовим сховищем для новостворених облікових записів комп’ютерів, забезпечуючи централізоване місце для цих об’єктів. Контейнер Domain Controllers спеціально розроблений для розміщення всіх облікових записів DC, гарантуючи їх організованість та легкий доступ. Контейнер ForeignSecurityPrincipals зарезервований для ідентифікаторів безпеки (SIDs) із зовнішніх доменів, полегшуючи міждоменну безпеку та дозволи. Контейнер Keys зберігає об’єкти криптографічних ключів, які є важливими для безпечних комунікацій та шифрування в мережі. Контейнер LostAndFound відіграє критичну роль у підтримці цілісності каталогу, утримуючи осиротілі об’єкти, які від’єдналися від своїх оригінальних контейнерів, що запобігає потенційним проблемам з посиланнями на об’єкти. Контейнер Managed Service Accounts присвячений керованим обліковим записам служб, які використовуються для забезпечення підвищеної безпеки та управління службами, що працюють на серверах. Контейнер Users є типовим місцем для оновлених або новостворених облікових записів користувачів, що полегшує управління та доступ до об’єктів, пов’язаних з користувачами [15].

Після встановлення розуміння цих типових контейнерів, наступним кроком є розгляд концепції делегування контролю організаційній одиниці (OU). Делегування контролю передбачає призначення конкретних адміністративних дозволів користувачам або групам для певних OU, що дозволяє їм керувати об’єктами в межах цієї OU без надання повних адміністративних прав у всьому середовищі AD. Цей процес делегування є критично важливим для підтримки безпечного та організованого каталогу, оскільки він дозволяє адміністраторам призначати обов’язки користувачам без прав адміністратора, обмежуючи їхній доступ лише тими об’єктами та функціями, які необхідні для їхніх ролей. Такий підхід допомагає збалансувати адміністративний контроль із безпекою, гарантуючи, що користувачі мають належний рівень доступу для ефективного виконання своїх завдань без компрометації цілісності загальної інфраструктури AD.

Розуміння функції OU в AD є необхідним для ефективного управління каталогом. OU слугують засобом для систематичної організації та управління об’єктами AD. Для підвищення ефективності адміністрування контроль може бути делеговано конкретним користувачам або групам у межах OU. Цей процес дозволяє розподіляти адміністративні обов’язки без надання користувачам повних адміністративних прав у всьому середовищі AD.

Для делегування контролю користувачі або групи спочатку повинні бути переміщені у визначену OU, як показано на рисунку 12.21.

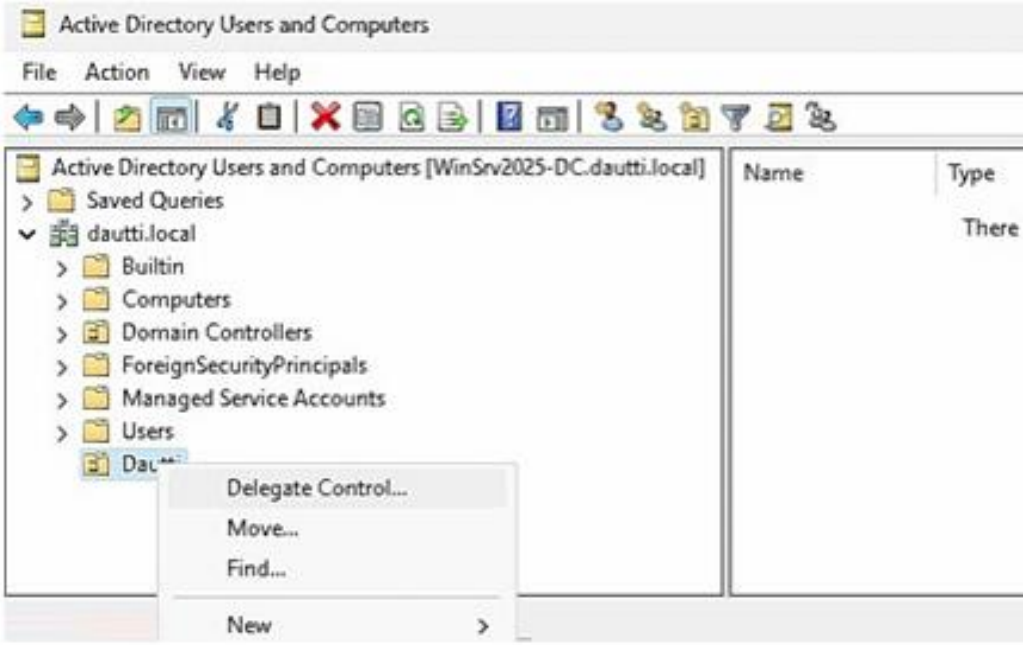

Рисунок 12.15 – Делегування керування організаційномій одиниці у Windows Server 2025 [13]

Делегування контролю включає призначення конкретних адміністративних дозволів, таких як управління обліковими записами користувачів, скидання паролів або зміна членства в групах у межах цієї OU. Це цілеспрямоване делегування допомагає забезпечити виконання адміністративних завдань відповідним персоналом, зберігаючи при цьому безпеку та організованість. Обмежуючи дозволи конкретними OU, адміністратори можуть ефективніше керувати ресурсами та знижувати ризик несанкціонованого доступу або ненавмисних змін. Делегування контролю також уможливлює впровадження адміністрування на основі ролей, що може покращити операційну ефективність та підзвітність. Кожному делегованому адміністратору можуть бути призначені завдання, що відповідають його ролі, що полегшує відстеження змін та управління об’єктами каталогу відповідно до організаційних політик. У наступному розділі буде детальніше розглянуто управління обліковими записами користувачів, обліковими записами комп’ютерів та групами в AD, досліджено, як ці елементи взаємодіють з OU та сприяють створенню добре структурованого та безпечного середовища каталогу.

Перемещаємі профіля і домашні каталоги

Розуміння різних типів профілів користувачів у середовищах Windows Server визначається як фундаментальне для ефективного управління користувачами та налаштування системи. Профіль користувача являє собою сукупність налаштувань реєстру, файлової структури та даних, що визначають робоче середовище конкретного користувача. У наступному викладі надається пояснення трьох основних типів профілів користувачів в Active Directory: локальних профілів, які прив’язані до конкретної машини; переміщуваних профілів, які пропонують гнучкість при роботі на кількох пристроях; та обов’язкових профілів, які підтримують фіксовану конфігурацію без можливості модифікації користувачем.

Локальний профіль користувача створюється автоматично, коли користувач виконує вхід у комп’ютер вперше. Цей профіль зберігається на жорсткому диску конкретної машини (зазвичай у директорії C:\Users), як зображено на рисунку 12.16.

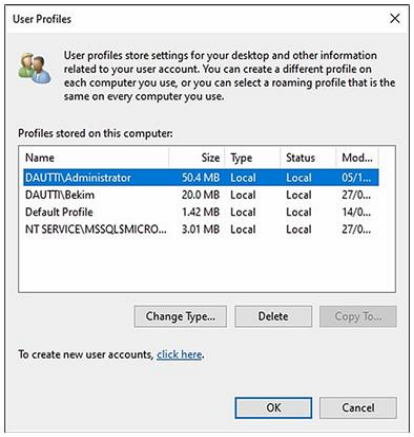

Рисунок 12.16 – Профілі користувачів у Windows Server 2025 [13]

Локальний профіль включає налаштування користувача (файл NTUSER.DAT), документи, а також дані програм, адаптовані до цього конкретного комп'ютера. Локальний профіль вважається ідеальним для індивідуального використання на одній машині, проте йому бракує гнучкості у випадках, коли користувачам необхідно отримувати доступ до свого робочого середовища з декількох пристроїв, оскільки налаштування не синхронізуються між різними робочими станціями [13].

Переміщуваний профіль користувача (Roaming User Profile) розширює гнучкість робочого середовища, дозволяючи користувачам отримувати доступ до своїх персоналізованих налаштувань та файлів з будь-якого комп'ютера в межах мережі. Цей профіль, по суті, є копією локального профілю, що зберігається на централізованому мережевому ресурсі (файловому сервері). Процес функціонування переміщуваного профілю реалізується наступним чином: під час входу користувача система завантажує профіль з сервера на локальний клієнтський комп'ютер; під час сеансу зміни вносяться локально; при виході з системи всі зміни синхронізуються назад на сервер. Це забезпечує послідовний досвід роботи на різних машинах. Такий тип профілю є особливо корисним у середовищах, де користувачі часто змінюють робочі місця, наприклад, у офісах з незакріпленими робочими місцями (hot-desking) [13].

Обов'язковий профіль користувача (Mandatory User Profile) забезпечує дотримання фіксованої конфігурації профілю. Ці профілі, які також зберігаються на мережевому ресурсі, базуються на попередньо налаштованому шаблоні. Ключова технічна відмінність полягає у зміні розширення файлу куща реєстру з .DAT на .MAN. Будь-які зміни, внесені користувачем під час сеансу (наприклад, зміна фонового малюнка або налаштувань робочого столу), не зберігаються після виходу з системи. Це гарантує, що кожного разу, коли користувач виконує вхід, він починає роботу з тією ж базовою конфігурацією. Такий підхід є корисним у середовищах, де вимагається уніфікованість і небажані налаштування користувача, наприклад, у навчальних класах, інтернет-кіосках або на публічних терміналах [13].

Підсумовуючи, локальні профілі прив'язані до окремих комп'ютерів, переміщувані профілі забезпечують мобільність завдяки доступності з будь-якої мережевої машини, а обов'язкові профілі підтримують узгодженість шляхом скасування змін користувача та використання фіксованого шаблону. Кожен тип профілю слугує окремим цілям, допомагаючи адміністраторам ефективно керувати середовищами користувачів відповідно до організаційних потреб.

Окремою, але тісно пов'язаною з профілями концепцією, є використання домашніх каталогів (Home Directories). У той час як переміщуваний профіль призначений для зберігання налаштувань середовища (App Data, налаштування робочого столу), домашній каталог використовується як централізоване місце для зберігання особистих файлів та документів користувача. Налаштування домашнього каталогу виконується через властивості облікового запису користувача в Active Directory. При вході в систему домашній каталог автоматично підключається як мережевий диск (наприклад, диск H:) у сеансі користувача. Використання домашніх каталогів дозволяє відокремити дані користувача від його профілю, що пришвидшує процес входу в систему (оскільки не потрібно завантажувати великі обсяги

документів разом із профілем) і спрощує централізоване резервне копіювання критично важливих даних на сервері. Це також забезпечує доступ до документів з будь-якого комп'ютера в мережі, навіть якщо переміщувані профілі не використовуються [15].

Поєднання переміщуваних профілів для налаштувань та домашніх каталогів для даних вважається найкращою практикою для забезпечення повноцінної мобільності користувачів у корпоративній мережі.

Безпека файлової системи і принципи побудови безпеки (користувачі, групи, права)

Безпека, побудована навколо Active Directory, була спроектована для захисту цінних мережевих активів. На розвиток безпеки Windows Server, починаючи з версії 2012, вплинула ініціатива Trustworthy Computing («Надійні обчислення») від Microsoft, яка змінила основний фокус продуктів компанії на безпеку. По суті, увага до безпеки продуктів є вищою, ніж будь-коли раніше, і всі нові функції повинні пройти так званий «лакмусовий тест» на безпеку перед випуском. Ця ініціатива вплинула на розробку серверних операційних систем і чітко простежується у функціях безпеки.

Важливим елементом системи безпеки є протокол автентифікації Kerberos. Спочатку він був розроблений у MIT як безпечний метод автентифікації користувачів без фактичного відправлення пароля користувача через мережу, незалежно від того, зашифрований він чи ні. Можливість передачі даних автентифікації таким чином значно знижує загрозу крадіжки пароля, оскільки зловмисники більше не можуть перехопити копію пароля під час його проходження через мережу та застосувати атаки грубої сили (brute-force) для його розшифровки. Фактична функціональність Kerberos є складною, але, по суті, відбувається наступне: комп'ютер надсилає клієнту інформаційний пакет, який вимагає автентифікації. Цей пакет містить свого роду «загадку», відповідь на яку може бути надана лише за допомогою правильних облікових даних користувача. Користувач застосовує «відповідь» до загадки та надсилає її назад на сервер. Якщо до відповіді було застосовано правильний пароль, користувач проходить автентифікацію. Хоча ця форма автентифікації використовується в Windows Server, вона не є власністю Microsoft і доступна як інтернет-стандарт.

Реалізації AD DS, по суті, є настільки безпечними, наскільки безпечним є середовище Windows Server, у якому вони працюють. Безпека структури AD DS може бути підвищена за допомогою додаткових запобіжних заходів, таких як захищений зв'язок між серверами з використанням IPsec або використання смарт-карт та інших методів шифрування. Крім того, середовище користувача може бути захищене за допомогою групових політик, які дозволяють встановлювати зміни параметрів, такі як обмеження паролів користувачів, безпека домену та привілеї доступу при вході в систему.

У Windows Server 2025 представлено комплексний набір покращень безпеки для AD DS, розроблений для посилення протоколів автентифікації та захисту критичної інфраструктури від кіберзагроз, що стають дедалі складнішими. В основі цих покращень лежить вдосконалення автентифікації Kerberos, яка протягом багатьох років була наріжним каменем AD DS. У цьому випуску Kerberos отримує переваги від більш надійних алгоритмів шифрування, що посилює захист від поширених вразливостей, таких як атаки Pass-the-Ticket та Golden Ticket. Ці вразливості часто використовуються для атак з бічним переміщенням всередині мережі. Завдяки цим оновленням Windows Server 2025 забезпечує дотримання вищих криптографічних стандартів, знижуючи ймовірність крадіжки облікових даних та несанкціонованого доступу [13].

Ще одним ключовим досягненням визначається покращена інтеграція можливостей багатофакторної автентифікації (MFA), які стали необхідними для пом'якшення сучасних ризиків кібербезпеки. Windows Server 2025 забезпечує більш плавну та гнучку інтеграцію MFA з політиками умовного доступу, що дозволяє організаціям динамічно застосовувати MFA на основі різних факторів, таких як чутливість ресурсів, до яких здійснюється доступ, роль користувача або навіть місцезнаходження доступу. Такий контекстний підхід до безпеки гарантує, що привілейовані ресурси отримують найвищий рівень захисту, мінімізуючи при цьому незручності для звичайних користувачів, які виконують рутинні завдання. Включення біометрії та MFA на основі токенів додатково покращує структуру безпеки, узгоджуючись із

моделями безпеки Zero Trust («Нульова довіра»), які надають пріоритет безперервній перевірці [13].

Управління груповою політикою також зазнало значних покращень у Windows Server 2025. Адміністратори тепер можуть впроваджувати та забезпечувати дотримання політик безпеки з підвищеною точністю, отримуючи вигоду від ширшого діапазону попередньо визначених шаблонів безпеки, адаптованих для різних організаційних потреб. Ці шаблони, приклад яких (файли ADMX) наведено на рисунку 12.17, допомагають спростити розгортання конфігурацій безпеки в середовищах AD DS.

У поєднанні з можливостями моніторингу в реальному часі та оповіщення, ці покращення дозволяють швидше виявляти та усувати потенційні загрози безпеці. Організації можуть точно налаштовувати параметри контролю доступу та відстежувати активність користувачів, тим самим мінімізуючи вектори атак у мережі.

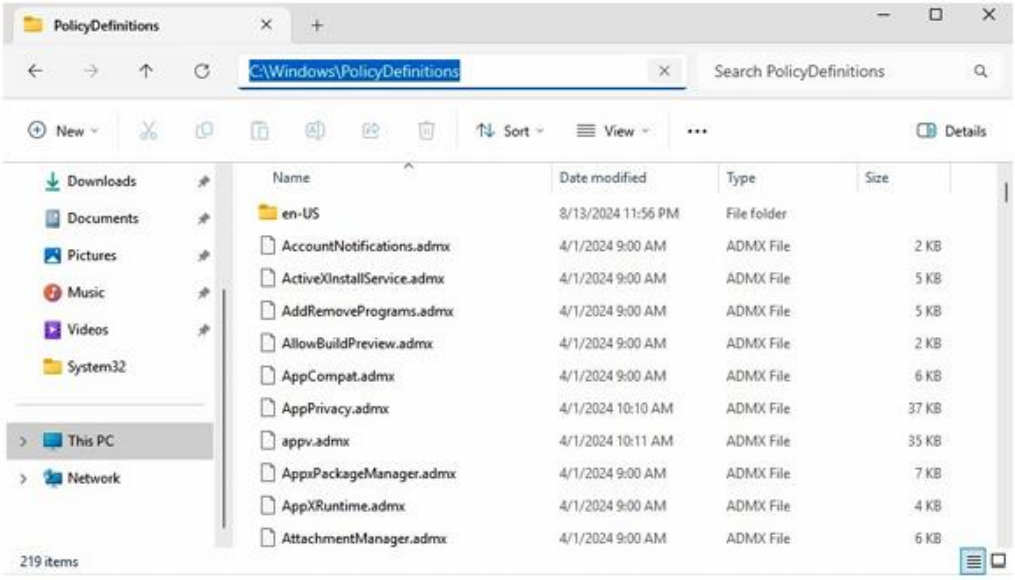

Рисунок 12.17 – Файли ADMX у Windows Server 2025 [13]

На додаток до цих функцій, Windows Server 2025 покращує механізми аудиту та логування в межах AD DS. Впровадження більш гранулярних можливостей аудиту дозволяє командам безпеки отримувати глибше розуміння патернів автентифікації та швидше виявляти аномалії. Завдяки покращеному логуванню подій організації можуть краще відстежувати спроби входу, патерни доступу та потенційні порушення, що є критично важливим для криміналістичних розслідувань та дотримання правил безпеки. Це гарантує, що команди безпеки обладнані для швидкого реагування на спроби несанкціонованого доступу та підтримки постійної пильності щодо мережі.

Окремо слід зазначити наявність великої кількості безкоштовних сценаріїв PowerShell у центрі скриптів Microsoft та галереї PowerShell. Ці платформи слугують відомими репозиторіями, де IT-фахівці можуть знайти та поділитися сценаріями для різних адміністративних завдань. Обидва ресурси містять великі колекції скриптів, що стосуються саме AD та DNS, що робить їх безцінними для автоматизації та спрощення складних заходів з управління мережею.

У сукупності ці досягнення в галузі безпеки позиціонують AD DS як життєво важливий інструмент для захисту сучасних IT-середовищ, особливо в організаціях із гібридними хмарними архітектурами. Зміцнюючи фундаментальні протоколи автентифікації, покращуючи інтеграцію MFA, вдосконалюючи управління політиками та забезпечуючи глибшу видимість подій безпеки, Windows Server 2025 пропонує комплексний захист від ландшафту загроз, що постійно розвивається. Ці покращення дозволяють організаціям прийняти проактивну позицію щодо безпеки, гарантуючи, що їхня IT-інфраструктура залишається стійкою до нових викликів кібербезпеки. Маючи чітке розуміння впроваджених розширених функцій безпеки та механізмів автентифікації, важливо розглянути, як ці

покращення інтегруються з ширшими IT-інфраструктурами.

Керування користувачами та групами в межах AD

Розуміння облікових записів користувачів і комп'ютерів, разом із групами, є фундаментальним для керування доступом до мережі в середовищі домену на базі Windows. Ці облікові записи є критично важливими елементами AD, що забезпечують автентифікацію як користувачів, так і пристроїв у всій мережі. У цій централізованій системі групи мають особливе значення, оскільки вони спрощують процес призначення та керування правами й дозволами. Групи агрегують численні облікові записи, дозволяючи адміністраторам застосовувати політики та дозволи колективно, а не індивідуально. Такий оптимізований підхід підвищує як безпеку, так і ефективність.

Розуміння облікових записів домену є необхідним для ефективного керування доступом до мережі в середовищі AD. Облікові записи домену проходять автентифікацію через AD, що дозволяє користувачам отримувати доступ як до локальних, так і до мережевих ресурсів відповідно до дозволів, призначених самому обліковому запису або успадкованих від членства в групах. Ця централізована структура автентифікації забезпечує оптимізований і безпечний підхід до керування доступом до різних служб і програм у мережі.

Для створення облікового запису домену у Windows Server 2025 виконується наступна послідовність дій: спочатку відкривається консоль «Active Directory – користувачі й комп'ютери» (Active Directory Users and Computers) шляхом переходу до інструментів Windows (Windows Tools). Далі виконується натискання правою кнопкою миші на контейнері Users, обирається пункт New, а потім – user. Після цього вводиться необхідна інформація про користувача, як показано на рисунку 12.18**,** і натискається кнопка Next [13].

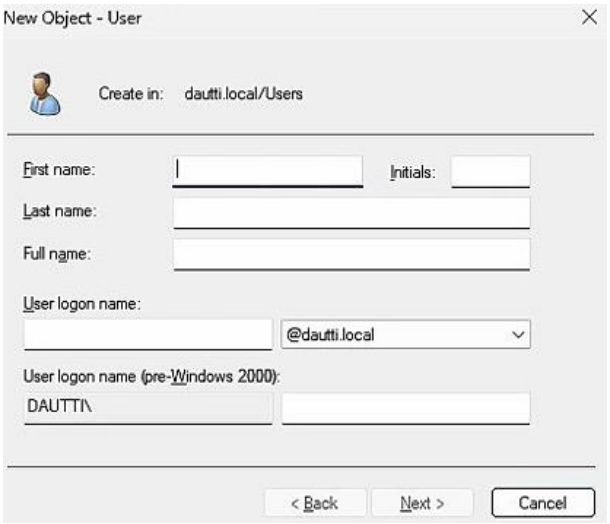

Рисунок 12.18 – Створення облікового запису домену в Windows Server 2025 [13]

На наступному етапі встановлюється тимчасовий пароль, підтверджується, і процес продовжується натисканням кнопки Next. Завершується створення облікового запису домену натисканням кнопки Finish. Цей процес встановлює обліковий запис домену та інтегрує його в структуру AD, надаючи користувачам доступ до мережевих ресурсів на основі призначених дозволів. У наступному підрозділі досліджується створення та керування локальними обліковими записами, які також є критичними для керування доступом користувачів та безпекою на окремих машинах і в межах специфічних локальних середовищ.

Розуміння локальних облікових записів є критично важливим для ефективного керування доступом на окремих комп'ютерах. На відміну від облікових записів домену, які проходять автентифікацію через AD і надають доступ у межах усієї мережі, локальні облікові записи є специфічними для комп'ютера, на якому вони створені, і керуються диспетчером безпеки облікових записів Windows (Security Accounts Manager – SAM). Ці облікові записи надають доступ до ресурсів на локальній машині та можуть взаємодіяти зі спільними ресурсами в одноранговій (P2P) мережі без необхідності отримання додаткових дозволів рівня домену. Локальні облікові записи є особливо корисними в сценаріях, коли комп'ютер

працює незалежно від домену або коли підключення до домену недоступне. Вони створюються та керуються локально, що дозволяє здійснювати гранулярний контроль над дозволами та доступом користувачів на основі кожної окремої машини. Це може бути вигідним для керування невеликими робочими групами або автономними комп'ютерами, де централізоване керування доменом є недоцільним [15].

Для створення локального облікового запису у Windows Server 2025 виконуються наступні кроки: здійснюється доступ до консолі «Керування комп'ютером» (Computer Management) через інструменти Windows (Windows Tools). Ця консоль надає централізований інтерфейс для керування різними компонентами системи, включаючи облікові записи користувачів. Потім необхідно перейти до System Tools, розгорнути розділ Local Users and Groups, натиснути правою кнопкою миші на контейнері Users і вибрати New, а потім – user. Далі вводяться необхідні дані користувача, такі як ім'я користувача та пароль, як зображено на рисунку 12.19. Процес завершується натисканням кнопки Create [13].

Важливим зауваженням при створенні локального облікового запису у Windows Server 2025 є те, що сервер не повинен функціонувати як контролер домену (DC). Якщо сервер призначено контролером домену, він оброблятиме функції, пов'язані з доменом, і керуватиме AD DS, що ускладнює керування локальними обліковими записами. Забезпечуючи, що сервер не є DC, можна уникнути складнощів керування доменом, що дозволяє просте налаштування та керування локальними обліковими записами без додаткового навантаження служб домену. Локальні облікові записи зберігаються та автентифікуються SAM на локальній машині, що гарантує дотримання контролю доступу та дозволів незалежно від домену мережі. Ці облікові записи ідеально підходять для сценаріїв, де потрібне локальне адміністрування та контроль доступу.

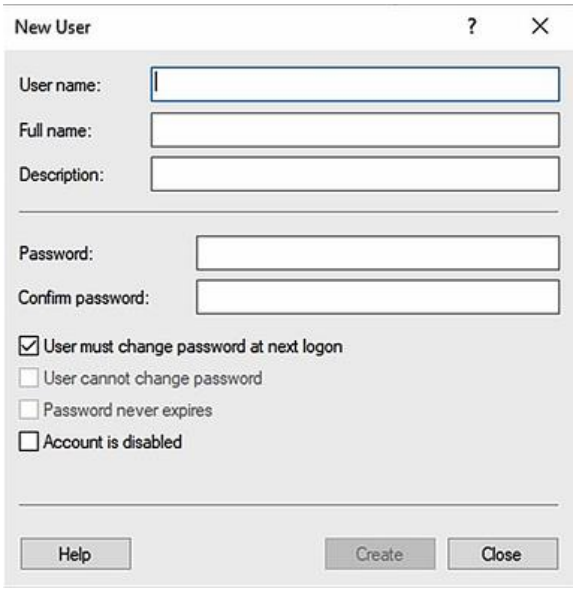

Рисунок 12.19 – Створення локального облікового запису в Windows Server 2025 [13]

У середовищі AD облікові записи комп'ютерів є критичними для ідентифікації та керування комп'ютерами в домені. Перед приєднанням до домену кожен комп'ютер повинен мати унікальне ім'я хоста для запобігання конфліктам. Цей унікальний ідентифікатор гарантує, що комп'ютер може бути точно відстежений і керований у мережі. Після успішного додавання комп'ютера до домену він зберігає своє ім'я хоста для постійної взаємодії з іншими ресурсами домену, включаючи файли, програми та служби. Це налаштування дозволяє безперебійну комунікацію та інтеграцію з доменом. Консоль «Active Directory – користувачі й комп'ютери» ефективно обробляє адміністрування облікових записів комп'ютерів, як показано на рисунку 12.20.

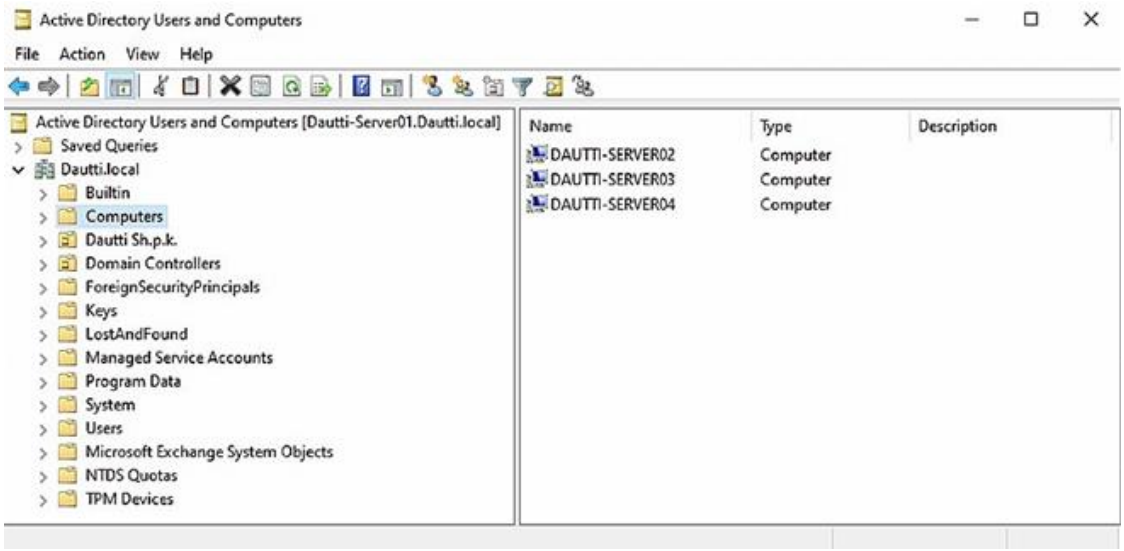

Рисунок 12.20 – Облікові записи комп’ютерів у Windows Server 2025 [13]

Ця консоль дозволяє адміністраторам переглядати та керувати обліковими записами комп’ютерів, налаштовувати властивості та застосовувати політики. Тут виконуються такі завдання, як скидання паролів, увімкнення або вимкнення облікових записів та зміна налаштувань облікового запису.

Розуміння облікових записів комп’ютерів є необхідним для підтримання цілісності мережі та забезпечення належного доступу до ресурсів. Ці облікові записи відіграють життєво важливу роль в автентифікації та авторизації комп’ютерів у домені, підтримуючи таким чином ефективне керування мережею та безпеку. Далі увага зміщується на групи в межах структури AD. Групи є невід’ємною частиною керування дозволами та правами доступу, спрощуючи призначення ролей і привілеїв та оптимізуючи адміністративні завдання. Вони допомагають організовувати користувачів і комп’ютери, застосовувати послідовні політики та підвищувати загальну безпеку мережі.

Розуміння типів груп в AD є фундаментальним для оптимізації керування мережею та забезпечення безпеки. Групи AD спрощують адміністрування дозволів і прав, дозволяючи адміністраторам керувати декількома об’єктами AD колективно, а не налаштовувати кожен об'єкт індивідуально. Такий підхід не тільки підвищує ефективність, але й допомагає підтримувати послідовні політики безпеки в мережі. Групи самі по собі також є об’єктами AD і можуть бути переміщені або реорганізовані в межах різних OU для узгодження з організаційними змінами або адміністративними потребами. Як представлено на рисунку 12.21, групи адмініструються за допомогою консолі «Active Directory – користувачі й комп’ютери». В AD групи класифікуються на дві основні категорії [13].

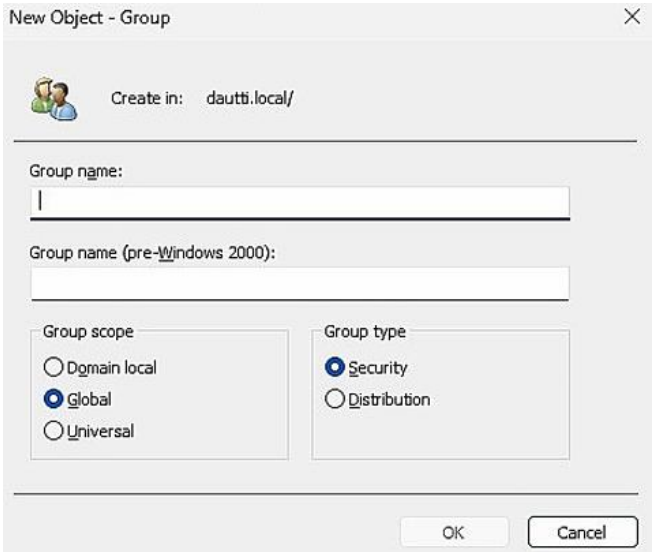

Рисунок 12.21 – Типи груп у Windows Server 2025 [13]

Групи безпеки (Security groups) – це групи, які є важливими для керування доступом до спільних мережевих ресурсів, таких як файли, папки та принтери. Вони застосовують дозволи та забезпечують дотримання політик безпеки в мережі. Групи безпеки можуть бути вкладені в інші групи безпеки для створення ієрархічної структури дозволів, що дозволяє здійснювати більш гранулярний контроль над доступом до ресурсів [13].

Групи розповсюдження (Distribution groups) – це групи, що спроектовані для полегшення розсилки електронних повідомлень в організації. Вони спрощують процес надсилання повідомлень великим групам користувачів, діючи як списки розсилки. Хоча групам розповсюдження не призначаються дозволи і вони не можуть використовуватися для контролю доступу до ресурсів, вони відіграють вирішальну роль у спрощенні внутрішньої комунікації [13].

Розуміння цих типів груп та їхніх функцій дозволяє адміністраторам ефективно керувати мережевими ресурсами та комунікацією. У наступних розділах будуть розглянуті типові групи – попередньо визначені групи, що постачаються з AD, – та процес створення нових груп. Це знання є необхідним для організації ролей користувачів, керування доступом до ресурсів та ефективного делегування адміністративних завдань у середовищі AD.

Розуміння типових груп (default groups) в AD є фундаментальним для ефективного адміністрування мережі. Коли сервер підвищується до ролі контролера домену (DC), він автоматично генерує різноманітні типові групи, як показано на рисунку 12.22, який ілюструє типові групи у Windows Server 2025. Ці типові групи спроектовані для спрощення адміністративних завдань шляхом групування пов'язаних об'єктів AD, тим самим полегшуючи процес призначення дозволів та прав доступу.

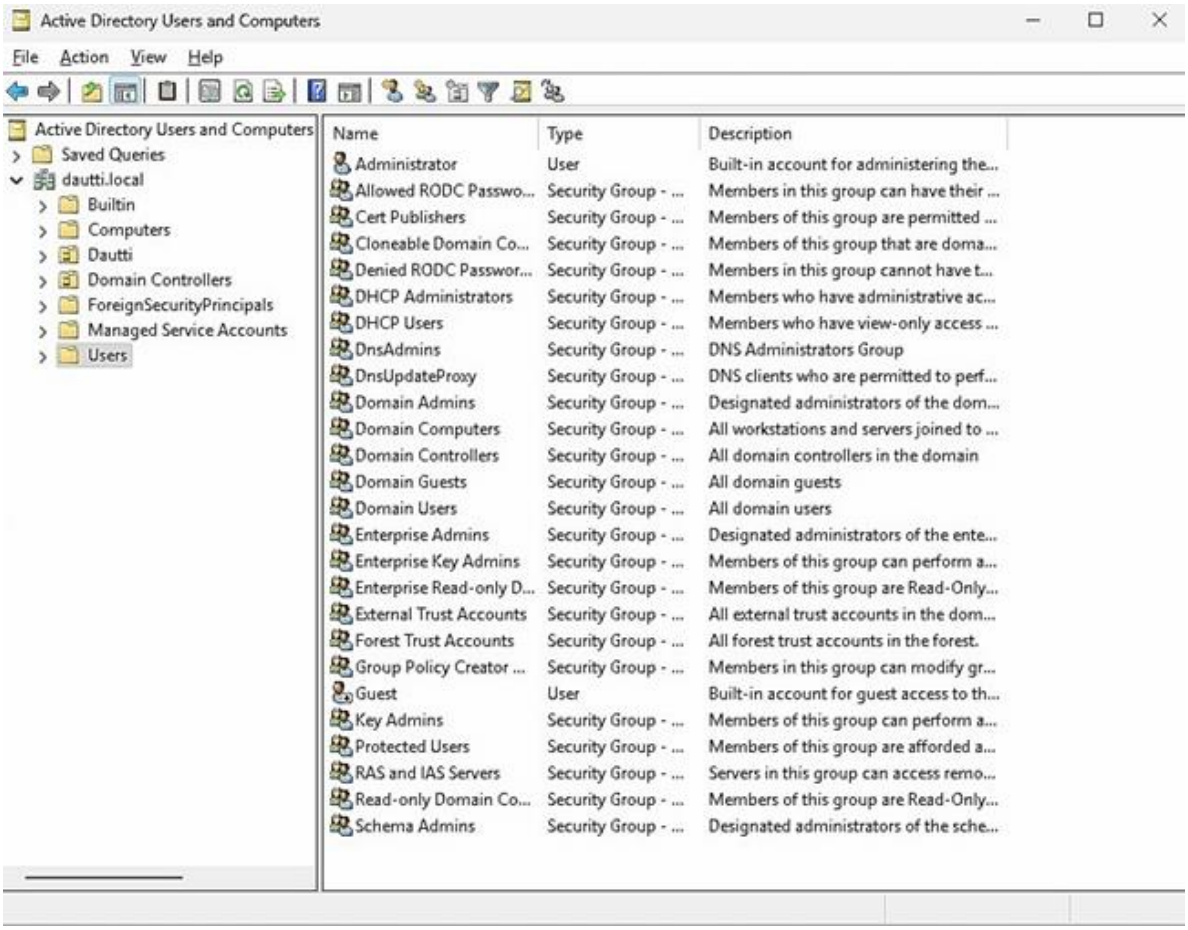

Рисунок 12.22 – Групи за замовчуванням у Windows Server 2025 [13]

Типові групи попередньо налаштовані зі специфічними ролями та дозволами, що може значно оптимізувати керування мережею. Наприклад, такі типові групи, як Domain Admins, Enterprise Admins та Schema Admins, мають попередньо визначені рівні адміністративних привілеїв, які є критичними для керування різними аспектами середовища

AD. Використовуючи ці групи, адміністратори можуть ефективно керувати доступом користувачів та забезпечувати дотримання політик безпеки без необхідності налаштовувати дозволи для кожного користувача або об'єкта вручну. Крім того, типові групи допомагають забезпечити послідовне застосування політик і дозволів у мережі, що підвищує як безпеку, так і операційну ефективність. Вони також полегшують делегування адміністративних завдань, надаючи попередньо визначені ролі, які можуть бути призначені користувачам залежно від їхніх обов'язків.

Розуміння областей дії груп є основоположним аспектом ефективного керування середовищами AD, оскільки вони безпосередньо впливають на те, як дозволи та політики застосовуються в мережі організації. Області дії груп визначають охоплення та застосовність членства в групах у структурі AD, що є критичним для підтримки безпеки та ефективності керування ресурсами. В AD існує три основні області дії груп, кожна з яких слугує відмінним цілям та контекстам, як показано на рисунку 12.23.

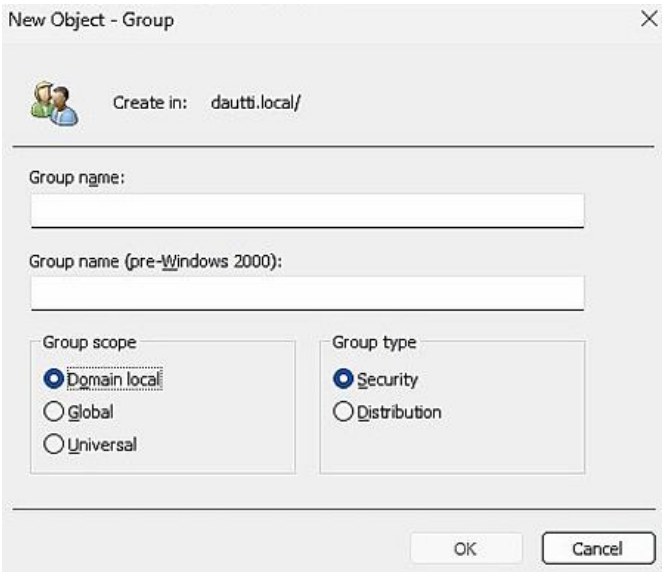

Рисунок 12.23 – Області дії груп у Windows Server 2025 [13]

Локальна група домену (Domain local group scope) – це область дії, яка спроектована для керування доступом до ресурсів у межах локального домену. Вона дозволяє включати облікові записи, локальні групи домену, глобальні групи та універсальні групи, що дозволяє адміністраторам ефективно призначати дозволи для локальних ресурсів. Локальні групи домену є особливо корисними при керуванні доступом до ресурсів, таких як файлові ресурси, принтери та інші специфічні для домену ресурси, де потрібно обмежити доступ користувачам і групам у межах цього домену [13].

Глобальна група (Global group scope) – це область дії, яка використовується для організації користувачів і груп у межах одного домену, які мають спільні вимоги до доступу. Ця область дії включає облікові записи та глобальні групи, специфічні для глобальної групи батьківського домену. Глобальні групи зазвичай використовуються для призначення дозволів на ресурси в різних доменах у межах одного лісу, що робить їх ідеальними для сценаріїв, де користувачі з кількох доменів потребують доступу до спільних ресурсів [13].

Універсальна область дії групи (Universal group scope) є найбільш широкою, дозволяючи включення облікових записів, глобальних груп та універсальних груп з будь-якого домену в лісі. Ця область дії є необхідною для керування дозволами в декількох доменах, що робить її високоефективною у великих багатодоменних середовищах. Універсальні групи є особливо корисними, коли потрібно призначити дозволи послідовно по всьому лісу, гарантуючи, що користувачі в різних доменах мають належний доступ до ресурсів незалежно від їхнього членства в домені.

Кожна з цих областей дії груп відіграє критичну роль у забезпеченні належного та послідовного застосування дозволів і політик у середовищі AD. Розуміючи та правильно використовуючи ці області дії, адміністратори можуть підвищити як ефективність, так і

безпеку своїх практик керування мережею. Крім того, належне використання областей дії груп може запобігти поширеним проблемам, таким як надмірні дозволи, коли користувачі мають більше доступу, ніж необхідно, або недостатні дозволи, коли законний доступ заборонено. Цей баланс є критичним для підтримки безпечного та добре функціонуючого середовища AD.

Концепція вкладеності груп базується на принципах областей дії груп, дозволяючи адміністраторам створювати більш складні та гнучкі структури груп. Вкладеність груп додатково вдосконалює здатність керувати дозволами та правами доступу, пропонуючи потужний інструмент для масштабних середовищ AD. Розуміння вкладеності груп в AD є фундаментальним аспектом ефективного та безпечного керування дозволами в складних IT-середовищах. Вкладеність груп дозволяє ієрархічно організовувати групи, що дає змогу адміністраторам ефективніше призначати дозволи, використовуючи структурований, багаторівневий підхід. Цей метод не тільки спрощує адміністрування контролю доступу, але й зменшує надлишковість та потенційні помилки, які можуть виникнути при індивідуальному призначенні дозволів численним обліковим записам користувачів.

На практиці вкладеність груп керується найкращими практиками, такими як методології Microsoft AGDLP (Accounts, Global, Domain Local, Permissions) та AGUDLP (Accounts, Global, Universal, Domain Local, Permissions). Ці моделі пропонують систематичний підхід до керування членством у групах та дозволами в мережі [13].

У моделі AGDLP облікові записи користувачів спочатку призначаються глобальній групі, яка зазвичай представляє певну роль або відділ в організації. Ця глобальна група потім вкладається в локальну групу домену, яка відповідає за керування доступом до конкретних ресурсів у межах локального домену. Дозволи призначаються локальній групі домену, тим самим надаючи доступ усім членам глобальної групи за один крок. Цей метод є особливо ефективним у середовищах, де користувачі потребують послідовного доступу до ресурсів у межах одного домену [13].

Методологія AGUDLP розширює модель AGDLP, включаючи універсальну групу в структуру вкладеності. Тут глобальна група спочатку додається до універсальної групи, яка може охоплювати кілька доменів у межах лісу. Потім універсальна група включається в локальну групу домену, яка контролює доступ до ресурсів. Цей підхід ідеально підходить для великих багатодоменних середовищ, де користувачі потребують доступу до ресурсів у різних доменах. Використовуючи універсальні групи, адміністратори можуть підтримувати послідовну структуру дозволів по всьому лісу, гарантуючи, що користувачі мають необхідний доступ незалежно від домену, в якому вони працюють [13].

Ці структуровані методології не лише оптимізують керування дозволами, але й підвищують безпеку та масштабованість середовища AD. Зменшуючи кількість індивідуальних призначень дозволів і централізуючи контроль у чітко визначених структурах груп, адміністратори можуть легше забезпечувати дотримання політик безпеки, проводити аудит контролю доступу та реагувати на організаційні зміни. Після отримання ґрунтовного розуміння фундаментальних елементів AD, таких як DNS, OU та контейнери, а також класифікації облікових записів комп’ютерів і груп, наступним кроком є перехід до встановлення ролей AD DS і DNS. Цей етап є критичним, оскільки він закладає основу для налаштування та керування середовищем AD, гарантуючи, що воно відповідає потребам організації щодо безпеки, масштабованості та адміністрування.

Створення доменів і лісів

Процес розгортання інфраструктури Active Directory Domain Services (AD DS) розпочинається зі створення логічної основи, ключовими елементами якої виступають домени та ліси. Створення нового лісу визначається як фундаментальний етап проектування мережевого середовища, оскільки саме ліс встановлює межі безпеки, реплікації та адміністрування для всіх об’єктів директорії.

Практична реалізація розгортання інфраструктури розпочинається з підготовки віртуалізованого середовища. Перед початком виконання завдань ініціюється запуск віртуальної машини під керуванням операційної системи Windows Server 2025, що функціонує в середовищі гіпервізора Hyper-V. Дана віртуальна машина має бути попередньо

налаштована в межах підготовчих етапів розгортання лабораторного стенду.

Після успішного завантаження операційної системи процес конфігурації виконується через централізовану консоль керування «Диспетчер серверів» (Server Manager). Для створення та налаштування домену першочерговим завданням є інсталяція відповідної ролі сервера. У меню «Управління» (Manage) обирається пункт «Додати ролі та компоненти» (Add Roles and Features), що ініціює запуск майстра встановлення. Серед переліку доступних ролей обирається пункт «Доменні служби Active Directory» (Active Directory Domain Services). Цей алгоритм є стандартизованим для серверних операційних систем сімейства Windows [15].

Після вибору необхідної ролі здійснюється перехід до наступних етапів майстра, що завершується вікном підтвердження налаштувань. На цьому етапі підтверджується інсталяція обраних компонентів, після чого розпочинається процес копіювання бінарних файлів та налаштування залежностей.

Завершення процесу встановлення ролі не означає автоматичного створення домену. Сервер потребує процедури підвищення статусу (promotion). Після закінчення інсталяції в області сповіщень «Диспетчера серверів» з'являється інтерактивне повідомлення «Підвищити роль цього сервера до рівня контролера домена» (Promote this server to a domain controller). Активація цього посилання ініціює запуск спеціалізованого інструменту конфігурації.

Після натиснення відкривається «Майстер налаштування доменних служб Active Directory» (Active Directory Domain Services Configuration Wizard). Цей етап є критичним, оскільки саме тут визначається топологія майбутньої мережі.

У вікні вибору операції розгортання адміністратору пропонується обрати один із трьох сценаріїв, кожен з яких відповідає певним архітектурним потребам.

«Додати контролер домена в існуючий домен» (Add a domain controller to an existing domain) – ця опція використовується для масштабування інфраструктури, коли домен вже функціонує, і виникає потреба у додатковому контролері для розподілу навантаження або забезпечення відмовостійкості.

«Додати новий домен в існуючий ліс» (Add a new domain to an existing forest) – обирається у випадку, коли організація вже має розгорнутий ліс і потребує створення дочірнього домену або нового дерева доменів у межах спільної інфраструктури.

«Додати новий ліс» (Add a new forest) – використовується при первинному розгортанні, коли створюється абсолютно нова інфраструктура, де відсутні як ліс, так і домени.

Оскільки в межах даного завдання створення домену виконується вперше, обирається опція «Додати новий ліс». У відповідному полі вводиться повне доменне ім'я (FQDN) кореневого домену, після чого здійснюється перехід до наступного етапу натисканням кнопки «Далі».

На етапі налаштування параметрів контролера домену визначаються функціональні рівні лісу та домену. Зазвичай ці параметри залишаються без змін, що відповідають версії поточної операційної системи. Критично важливим кроком є встановлення складного пароля для режиму відновлення служб каталогів (Directory Services Restore Mode – DSRM). Цей пароль використовується для аварійного відновлення бази даних Active Directory у випадку серйозних збоїв.

Подальші кроки налаштування, такі як «Параметри DNS» (DNS Options) та «Додаткові параметри» (Additional Options), у базовому сценарії розгортання не потребують модифікації, тому підтверджуються натисканням кнопки «Далі». Аналогічний підхід застосовується на вкладці «Шляхи» (Paths), де визначаються місця розташування бази даних NTDS, файлів журналів та папки SYSVOL. Рекомендується залишити значення шляхів за замовчуванням [15].

На етапі «Переглянути параметри» (Review Options) здійснюється фінальна верифікація введених даних. Після перевірки система автоматично проводить аналіз відповідності сервера попереднім вимогам. У разі успішного проходження перевірки ініціюється процес інсталяції. Завершення процедури просування сервера до ролі контролера

домену супроводжується автоматичним перезавантаженням операційної системи.

Після перезавантаження сервера верифікація успішності розгортання виконується через «Диспетчер серверів». У меню «Інструменти» (Tools) обирається консоль «Користувачі і комп'ютери Active Directory» (Active Directory Users and Computers). Відкриття вікна консолі та наявність у ньому створеного домену (наприклад, server.lab) з відповідною ієрархією контейнерів свідчить про те, що домен успішно створено, а сервер набув статусу контролера домену. На цьому етап створення та базового налаштування домену вважається завершеним [13].

Тема 13
Групові політики

Розуміння основ групових політик (GP) у Windows Server

У практиці системного адміністрування часто виникає необхідність примусового застосування специфічних конфігурацій у мережі організації з метою забезпечення узгодженості та безпеки. До таких завдань, наприклад, належать встановлення веб-сайту компанії як домашньої сторінки за замовчуванням у всіх браузерах на комп'ютерах організації, а також обмеження доступу до знімних носіїв інформації. Додатково може виникнути потреба у відключенні використання облікових записів Microsoft у системах Windows 10 та 11. Ці всі завдання ефективно вирішуються за допомогою групових політик (GP) у середовищі Windows Server, що забезпечує централізований спосіб застосування та примусового виконання політик без необхідності покладатися на інструменти або утиліти сторонніх розробників.

Групова політика – це критично важлива функція Windows Server, що дозволяє адміністраторам примусово застосовувати політики як на рівні користувача, так і на рівні комп'ютера [13].

За допомогою об'єктів групової політики (GPO) адміністратори мають можливість визначати та впроваджувати налаштування, що контролюють поведінку користувачів та комп'ютерів у мережі. GPO надають адміністративні шаблони, які специфікують дозволені дії та конфігурації для користувачів і пристроїв, гарантуючи уніфіковане застосування організаційних стандартів та заходів безпеки. Окрім того, GPO можуть використовуватися як механізм безпеки для застосування критично важливих налаштувань захисту до користувачів та комп'ютерів у мережі під керуванням домену, що підвищує загальний рівень захищеності організації [24].

Важливо провести розмежування між перевагами групової політики (Group Policy Preferences, GPP) та налаштуваннями групової політики (Group Policy Settings, GPS). У той час як GPS примусово задають специфічні конфігурації та параметри для користувачів і комп'ютерів, GPP забезпечують більшу гнучкість, дозволяючи користувачам модифікувати свої налаштування без адміністративного втручання. Розуміння цієї відмінності сприяє ефективному використанню обох інструментів для керування середовищем.

За замовчуванням об'єкти групової політики зберігаються в директорії C:\Windows\SYSVOL\sysvol\<domain name>\Policies на контролері домену, як це продемонстровано на рисунку 13.1. Таке розташування за замовчуванням гарантує систематичне керування всіма налаштованими політиками та їх реплікацію між контролерами домену [13].

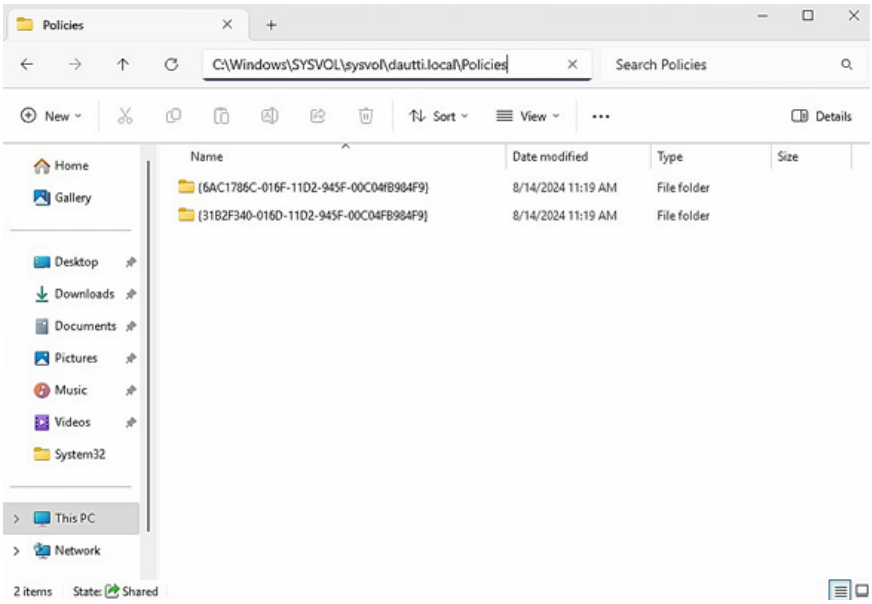

Рисунок 13.1 – Розташування GPO за замовчуванням у Windows Server 2025 [13]

Після формування ґрунтовного розуміння групових політик (GP) та об’єктів групових політик (GPO), важливо вивчити методи ефективного керування та конфігурації цих політик для узгодження з організаційними потребами та вимогами безпеки.

Варто зазначити, що ефективне усунення несправностей групових політик є критичним для підтримання належного функціонування середовища Windows Server. Використання таких інструментів, як «Результуюча політика» (Resultant Set of Policy, RSoP), gpresult та засіб перегляду журналів групової політики (Group Policy Log Viewer), здатне суттєво покращити можливості діагностики та вирішення проблем. Ці інструменти надають цінну інформацію про застосування політик, дозволяючи адміністраторам ідентифікувати конфлікти, оцінювати пріоритетність політик та гарантувати, що конфігурації застосовуються належним чином. У наступних розділах ці інструменти будуть детально розглянуті, що забезпечить оволодіння практичними стратегіями ефективного усунення несправностей в організації [25].

Групові політики та керування політиками Windows Server

Ефективне керування об’єктами групової політики (GPO) є критично важливим для системних адміністраторів з метою забезпечення примусового виконання та стандартизації конфігурацій у мережі. Основним інструментом для реалізації цього завдання виступає Консоль керування груповою політикою (Group Policy Management Console, GPMC), що пропонує централізований інтерфейс для створення, налаштування та застосування GPO у мережі на базі домену.

Інтерфейс GPMC розділено на дві основні панелі: панель лісу (Forest pane) та панель об’єктів GPO (GPOs pane). Панель лісу відображає ієрархічну структуру домену, надаючи адміністраторам можливість ефективної навігації між доменами та організаційними одиницями (OU). Панель GPO містить детальні вкладки, зокрема «Статус» (Status), «Пов’язані об’єкти групової політики» (Linked Group Policy Objects), «Спадкування групової політики» (Group Policy Inheritance) та «Делегування» (Delegation). Ці вкладки дозволяють переглядати поточний стан GPO, розуміти механізми застосування та спадкування політик, а також керувати налаштуваннями делегування для контролю повноважень щодо модифікації політик. На рисунку 13.2 продемонстровано інтерфейс GPMC на контролері домену з виділенням ключових компонентів та макета [13].

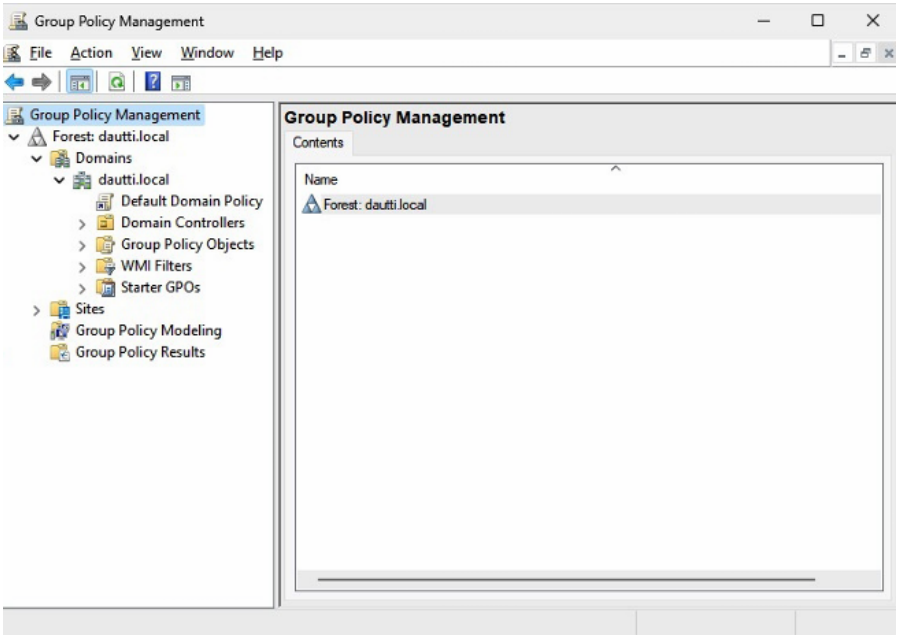

Рисунок 13.2 – Консоль GPM на контролері домену [13]

Доступ до консолі GPM у Windows Server 2025 може бути реалізований кількома методами, кожен з яких пропонує різні шляхи запуску інструменту залежно від адміністративних потреб. Ці методи, які забезпечують всебічне розуміння ефективного використання GPMC для керування налаштуваннями GP та гарантування послідовного

застосування політик у мережі, будуть розглянуті нижче. Важливо також зазначити необхідність надання чітких інструкцій щодо виключення або фільтрації конкретних користувачів та пристроїв із процесу застосування GP, оскільки багато середовищ вимагають адаптованих конфігурацій. Розуміння нюансів обробки GP, включаючи налаштування фільтрації та безпеки, дозволяє адміністраторам впевнено орієнтуватися у складних середовищах [15].

Адміністративні шаблони є критичним компонентом керування GP у Windows Server 2025, забезпечуючи структурований спосіб конфігурації та примусового застосування політик у середовищі. Шаблони для Windows Server постачаються у вигляді файлів формату .ADMX і зазвичай входять до складу інсталяції Windows Server. Однак для доступу до найновіших налаштувань, особливо для нових функцій, адміністраторам рекомендується періодично завантажувати актуальні версії з Центру завантажень Microsoft. Отримані файли .ADMX розміщуються у Центральному сховищі для GP у папці SYSVOL контролера домену, що гарантує можливість посилання всіх GPO на найновіші налаштування адміністративних шаблонів [13].

Процедура оновлення адміністративних шаблонів є важливою, оскільки Microsoft випускає нові шаблони з оновленнями та пакетами оновлень. Для оновлення виконуються наступні кроки: завантаження останніх версій з веб-сайту Microsoft, заміна існуючих файлів .ADMX у Центральному сховищі новими версіями, оновлення пов’язаних мовних файлів .ADML (які також зберігаються в Центральному сховищі) для відображення нових або змінених налаштувань, та оновлення всіх екземплярів GPMC для розпізнавання змін. Ефективне керування шаблонами вимагає дотримання найкращих практик, таких як регулярна перевірка оновлень (особливо після значних оновлень Windows), тестування нових шаблонів у невиробничому середовищі перед широким розгортанням для оцінки їх впливу, а також ведення документації щодо внесених змін, включаючи обґрунтування та дату оновлень, для сприяння майбутнім аудитам та усуненню несправностей. Впровадження цих практик дозволяє використовувати повний функціонал GP та мінімізувати ризики помилок конфігурації [26].

Для відкриття консолі GPM можна скористатися опцією «Засоби адміністрування» (Administrative Tools) у меню «Пуск», яка надає доступ до різноманітних інструментів керування. Процес розпочинається натисканням кнопки «Пуск» та переходом до розділу «Засоби Windows» (Windows Tools), де у списку наявних інструментів обирається «Керування груповою політикою» (Group Policy Management), як це проілюстровано на рисунку 13.3.

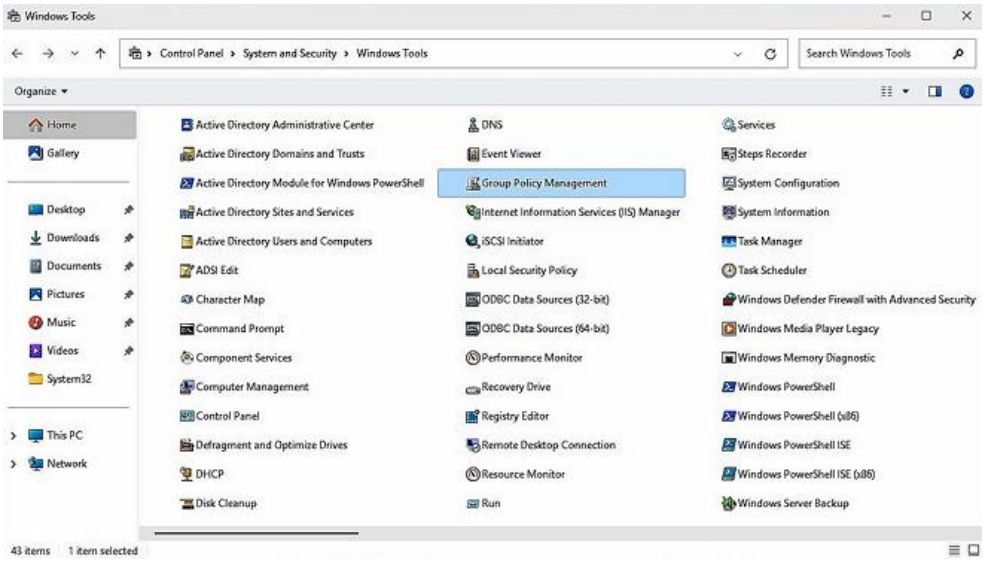

Рисунок 13.3 – Доступ до консолі GPM із засобів Windows [13]

Альтернативним та ефективним методом є використання діалогового вікна «Виконати», що дозволяє швидко отримати доступ до консолі за допомогою простої команди.

Для цього необхідно одночасно натиснути клавіші Windows + R, у текстовому полі ввести команду gpmc.msc та натиснути OK, як показано на рисунку 13.4. Ця команда забезпечує миттєвий запуск GPMC, надаючи централізований інтерфейс для керування налаштуваннями групових політик.

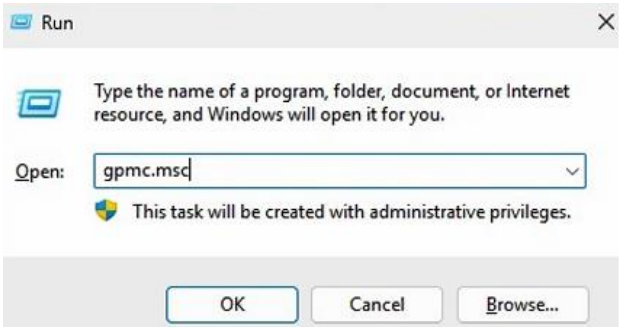

Рисунок 13.4 – Доступ до консолі GPM з діалогового вікна «Виконати» [13]

Також консоль GPM можна запустити через «Диспетчер серверів» (Server Manager) у меню «Пуск». Після відкриття вікна диспетчера серверів слід перейти до меню «Засоби» (Tools) та обрати «Керування груповою політикою», як зображено на рисунку 13.5. Ця дія відкриває GPMC, дозволяючи ефективно керувати налаштуваннями групових політик.

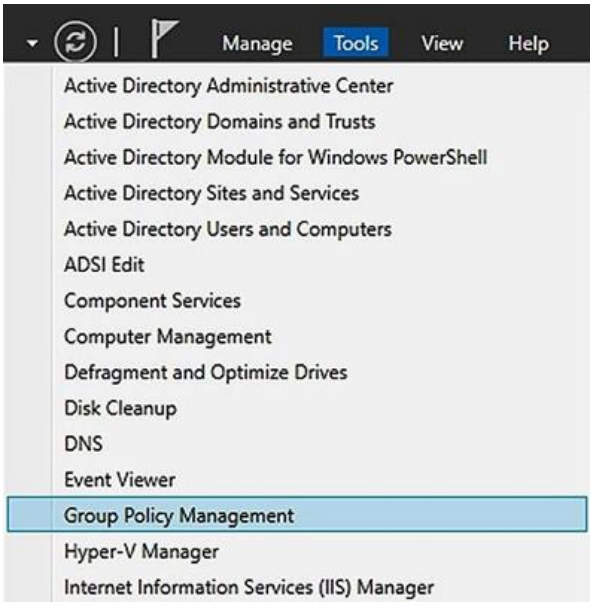

Рисунок 13.5 – Доступ до консолі GPM з диспетчера серверів [13]

Ефективне керування груповими політиками є необхідною умовою підтримання безпечного та ефективного середовища Windows Server. Дотримання найкращих практик дозволяє оптимізувати впровадження GP, забезпечити відповідність вимогам та мінімізувати потенційні проблеми. Рекомендується обмежувати використання стандартних політик (Default GPs), оскільки вони можуть бути надто складними. Натомість доцільно створювати спеціальні GPO, адаптовані до конкретних потреб організації, що зменшує ризик ненавмисних наслідків. Необхідно впровадити чітку конвенцію найменування GPO, що відображає їх призначення та сферу дії (наприклад, із префіксом відділу), що спрощує ідентифікацію та аудит. Важливим є регулярний перегляд та очищення застарілих або надлишкових GPO для підтримки керованості ландшафту політик та покращення продуктивності системи. Перед розгортанням у виробничих системах нові або модифіковані GPO завжди повинні проходити тестування у контрольованому середовищі для виявлення конфліктів. Адміністратори зобов’язані вести ретельну документацію налаштувань, призначень та змін GPO. Крім того, слід використовувати фільтрацію безпеки та фільтрацію інструментарію керування Windows (WMI Filtering) для цільового застосування GPO до

конкретних користувачів або груп, а також використовувати функцію моделювання GP (GP Modeling) для прогнозування впливу політик перед їх впровадженням [26].

Групова політика відіграє ключову роль у підвищенні організаційної ефективності та безпеки, узгоджуючи IT-практики з бізнес-цілями. Розглянемо реальні сценарії застосування. У фінансовій установі середнього розміру для зниження ризиків було впроваджено GPO для примусового виконання суворих політик паролів (зміна кожні 90 днів) та відключення локальних адміністративних прав на робочих станціях, що призвело до зниження кількості інцидентів безпеки на 40% протягом першого року. В університетському середовищі використання GP дозволило стандартизувати налаштування робочих столів, конфігурації принтерів та доступу до мережевих дисків, що зменшило кількість звернень до служби підтримки щодо проблем конфігурації на 30% та покращило робочий процес викладачів і студентів. У медичній організації IT-команда використала GP для налаштування служб оновлення Windows Server (WSUS), забезпечуючи автоматичне отримання пристроями останніх оновлень та патчів безпеки. Цей проактивний підхід мінімізував час простою через вразливості програмного забезпечення та дозволив досягти високого рівня відповідності стандартам охорони здоров'я. Ці приклади підкреслюють можливості GP щодо посилення безпеки та покращення взаємодії з користувачем у різних контекстах. У наступному розділі будуть детально розглянуті різноманітні налаштування конфігурації GPO [13].

Обробка групових політик, безпека і примінення по групах: як це працює?

Першочерговим аспектом, який необхідно усвідомити адміністратору стосовно групових політик, є той факт, що політики обробляються та застосовуються комп'ютерами та користувачами. Обробка виконується у чітко визначені моменти: під час запуску комп'ютера, при вході користувача, а також через фіксовані періоди часу. У процесі цієї обробки для визначення доцільності застосування політики враховується множина факторів. Одним із таких важливих факторів є те, чи застосовувалася політика раніше, і якщо так, то чи зазнала вона змін з моменту останнього застосування. Цей та багато інших факторів використовуються для перевірки кожної політики перед її безпосереднім застосуванням до комп'ютера або користувача.

Комп'ютер здійснює обробку політик під час процедур запуску та зупинки, а також у ході періодичних фонових оновлень. За замовчуванням на рядових серверах та робочих станціях інтервал оновлень встановлено на рівні 90 хвилин із часовим зсувом від 0 до 30 хвилин. На контролерах доменів групові політики оновлюються кожні 5 хвилин. Впровадження зсуву є необхідним заходом для уникнення одночасної обробки або оновлення групових політик усіма комп'ютерами домену, що могло б призвести до зниження продуктивності контролерів доменів та рядових комп'ютерів. Якщо під час запуску комп'ютер здатен успішно виявити контролер домену з можливістю автентифікації та встановити з ним зв'язок, виконується обробка GPO. У ході цього процесу система перевіряє для кожного пов'язаного або успадкованого GPO, чи не змінилася політика з часу останнього циклу обробки, та чи виникає необхідність виконання стартових сценаріїв і перевірки інших вимог для повторного застосування політики. Під час зупинки системи та періодичних оновлень об'єкти групових політик знову підлягають перевірці на наявність оновлень або змін з моменту останнього застосування. Обробка GPO комп'ютера детермінується зв'язками GPO, фільтрами доступу та фільтрами інструментальних засобів керування Windows (Windows Management Instrumentation – WMI) [13].

Процес обробки GPO користувачів є значною мірою подібним до обробки GPO комп'ютерів. Ключова відмінність полягає в тому, що обробка GPO користувачів ініціюється при вході та виході користувача із системи, а також здійснюється періодично. Інтервал оновлення за замовчуванням для обробки GPO користувачів також дорівнює 90 хвилинам із додаванням зсуву від 0 до 30 хвилин. Обробка GPO користувачів визначається наявними зв'язками GPO та налаштованими фільтрами доступу.

Служба визначення розташування в мережі (Network Location Awareness – NLA) являє собою вбудовану у Windows службу, яка спроектована для визначення статусу підключення комп'ютера до інфраструктури Active Directory. Інфраструктура групових політик використовує NLA для прийняття рішення щодо необхідності завантаження та застосування

об’єктів групових політик. Ця функція групових політик використовується також при перевірці зв’язності мережі – так званому визначенні повільних каналів зв’язку.

У попередніх версіях операційних систем при обробці групових політик для визначення надійності мережі застосовувалося повільне виявлення зв’язків. Цей механізм використовував для перевірки зв’язності протокол ICMP (Internet Control Message Protocol) або пінгування, що характеризувалося недостатньою надійністю. Внаслідок цього обробка групових політик на мобільних та віддалених клієнтських робочих станціях була вкрай нестабільною. Коли робоча станція мобільного клієнта підключалася до корпоративної мережі через VPN-з’єднання або після виходу з режиму очікування чи сну, зміна у мережевій зв’язності зазвичай залишалася непоміченою системою, і об’єкти GPO не застосовувалися та/або не оновлювалися. У таких випадках єдиним способом застосування GPO до цих клієнтів було ручне примусове оновлення групових політик з командного рядка або перезавантаження комп’ютерів при підключенні до корпоративної мережі через провідні з’єднання Ethernet [15].

Починаючи з Windows Vista та у більш пізніх версіях, включаючи Windows Server 2012, 2016, 2022 та 2025, обробка групових політик використовує для виявлення змін у мережі перероблену службу NLA. Оновлена служба NLA значно ефективніше розпізнає зміни у мережевих підключеннях, і при встановленні з’єднання NLA перевіряє доступність контролера домену. За наявності зв’язку з контролером домену служба NLA повідомляє службу групових політик на комп’ютері, яка, у свою чергу, ініціює обробку параметрів групових політик як для комп’ютера, так і для користувача. Служба NLA функціонує незалежно від ICMP або ping, що саме по собі робить її більш надійною. Служба NLA повинна коректно виконуватися у більшості мереж без необхідності спеціальних налаштувань на мережевих пристроях або брандмауерах, навіть якщо протокол ICMP відключено або заблоковано.

Групові політики класифікуються на політики та переваги, а в їх структурі наявні різні частини, розділи та визначення, що регламентують порядок, час та умови обробки цих розділів. Обробка групових політик на стороні клієнта керується розширеннями групових політик на стороні клієнта (Client-Side Extension – CSE). Ці розширення представляють собою файли бібліотек DLL, які встановлюються в операційній системі сервера або локальної робочої станції. При обробці групових політик застосування налаштувань до клієнта відбувається саме за допомогою CSE. У клієнтських системах можуть існувати CSE як для політик, так і для переваг, і кожне розширення CSE має власну поведінку обробки [28].

Локальні групові політики (комп’ютер і користувач)

До Windows-систем та облікових записів користувачів Windows-систем можуть застосовуватися два різних типи політик: локальні групові політики та групові політики доменів Active Directory. Слід зазначити, що локальні групові політики існують у всіх Windows-системах, тоді як доменні групові політики доступні виключно в лісі Active Directory.

До випуску операційних систем Windows Vista та Windows Server 2008 сервери та робочі станції мали можливість містити та застосовувати лише одну політику локального комп’ютера та користувача. Ця політика містила параметри, які можна було застосувати до локального комп’ютера та об’єктів користувачів з метою управління безпекою та конфігураційними налаштуваннями [15].

У багатьох середовищах, зазвичай через наявність вимог застарілих або виробничих систем, кінцевим користувачам часто призначалося членство в групі локальних адміністраторів для робочих станцій. Це виключало застосування багатьох параметрів безпеки, які повинні були поширюватися як на локальні, так і на групові політики. Кінцеві користувачі, які входять до групи локальних адміністраторів, могли перекривати параметри та змінювати налаштування, що створювало загрозу компрометації безпеки або, як це траплялося частіше, призводило до зниження надійності системи (рис. 13.6).

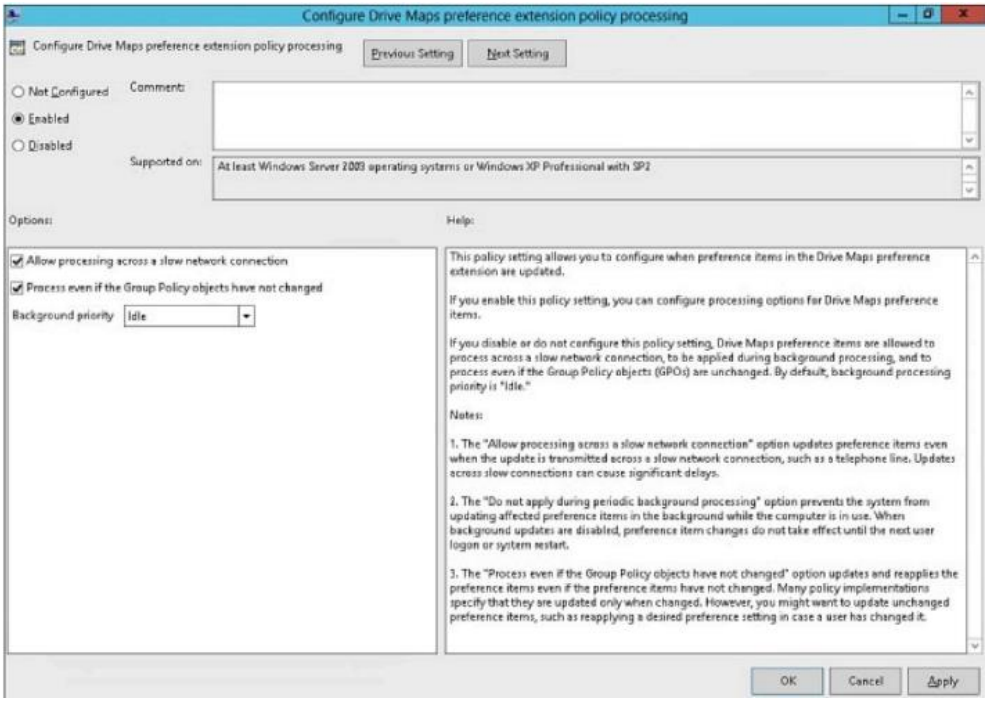

Рисунок 13.6 – Параметри обробки відображення дисків [15]

Крім того, політики можуть застосовуватися як до адміністраторів, так і до не адміністраторів локальних комп'ютерів. Це дозволяє адміністратору робочої станції нічого не вказувати в розділі користувача стандартної політики локального комп'ютера, натомість створюючи більш жорсткі політики для локальних користувачів і менш жорсткі – для членів групи адміністраторів на локальній робочій станції. Локальні групові політики користувачів можуть бути створені для конкретних користувачів, для всіх користувачів, які не є адміністраторами, а також для адміністраторів. Такий підхід дозволяє створювати різні конфігурації користувачів на основі облікових даних цих користувачів, зазначених під час входу в систему (рис. 13.7).

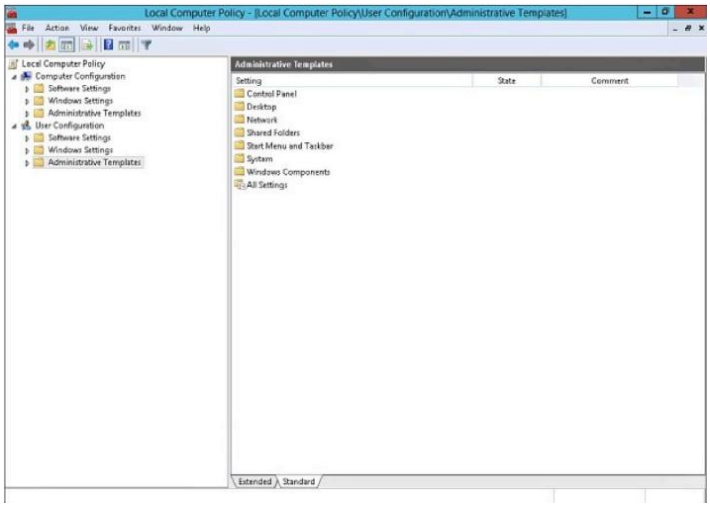

Рис. 13.7 – Перегляд параметрів локальної політики комп'ютера [15]

Групові політики на основі домену

Доменні групові політики мають суттєві відмінності від локальних групових політик, що зумовлено необхідністю наявності середовища Active Directory для їх створення та подальшого застосування. Ще однією значною відмінністю є те, що налаштування у групових політиках містять вузли як політик, так і вподобань, тоді як локальні групові політики позбавлені параметрів вподобань. Незважаючи на зазначені відмінності, більшість налаштувань залишаються ідентичними [15].

Слід зазначити, що доменні групові політики є більш зручними при визначенні

критеріїв, які використовуються для застосування політики. Передбачено можливість фільтрації доменних політик для їх застосування до конкретних членів груп безпеки Active Directory, комп'ютерів або об'єктів, розташованих у конкретній підмережі чи організаційній одиниці (OU). Також існує можливість застосування політик до комп'ютерів, що функціонують під управлінням конкретної версії операційної системи. Крім того, при визначенні вподобань у доменній груповій політиці можна вказувати застосовність параметрів на рівні окремих елементів, базуючись на різних критеріях (рис. 13.8).

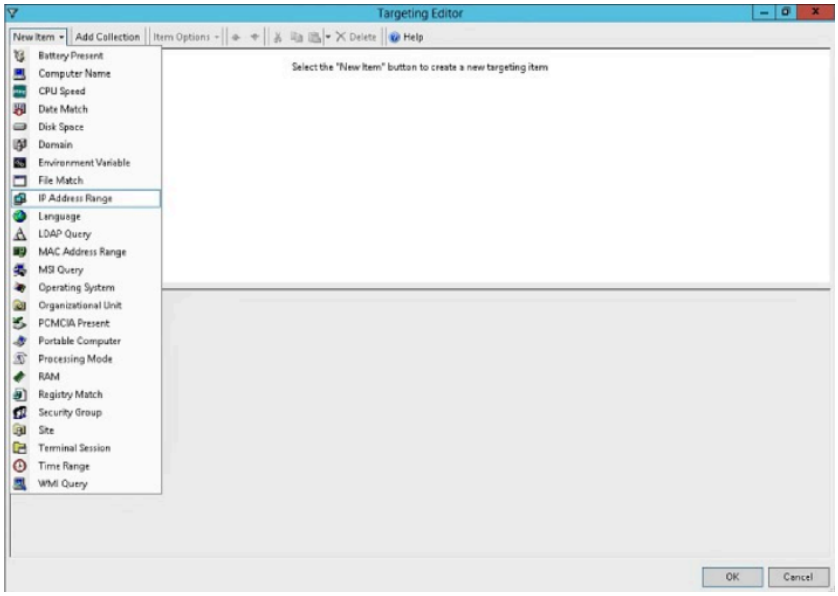

Рисунок 13.8 – Вказівка на рівні елементів для доменних GPP [15]

У структурі кожної політики локального комп'ютера та у вузлі конфігурації комп'ютера об'єкта групової політики (GPO) наявний розділ «Security Settings» (Параметри безпеки). Цей розділ включає параметри для політик аудиту комп'ютера, параметри управління обліковими записами, а також права, призначені користувачам. Унікальність цього розділу політики полягає в тому, що його можна імпортувати та експортувати окремо.

У попередніх версіях операційної системи Windows шаблони безпеки були сформовані заздалегідь, що надавало адміністраторам можливість швидкого завантаження набору рекомендованих параметрів конфігурації доступу. До переліку таких шаблонів належали базові шаблони робочої станції та сервера, а також шаблони високої безпеки, сумісної безпеки та безпеки контролера домену [15].

Для забезпечення управління стандартним набором налаштувань безпеки та його застосування до робочих груп і відокремлених систем адміністратори мають можливість скористатися функціями управління шаблонами безпеки. Використовуючи редактор об'єктів групових політик, редактор локальних політик безпеки або консоль налаштування та аналізу безпеки, можна імпортувати певний базовий шаблон, налаштувати або скоригувати параметри відповідно до наявних вимог, після чого експортувати чи зберегти ці параметри у спеціалізованому файлі шаблону. Надалі цей спеціалізований файл шаблону може бути імпортований або застосований до всіх необхідних систем за допомогою вищезгаданих інструментів.

Варто зауважити, що шаблони безпеки існують у Windows Vista, Windows Server 2008 та новіших версіях Windows. Ці базові шаблони безпеки розміщуються в каталозі %systemroot%\inf або c:\windows\inf. Номенклатура імен усіх стандартних шаблонів безпеки передбачає початок з deflt та розширення .inf. Наприклад, у Windows Server 2012 наявні шаблони з іменами defltbase.inf, defltsv.inf та defltdc.inf, за допомогою яких можна виконати стандартну конфігурацію параметрів безпеки системи [15].

Окрему увагу слід приділити ризикам, пов'язаним із застосуванням шаблонів. Імпорт шаблонів безпеки на вже розгорнуті сервери, робочі станції та контролери доменів може призвести до виникнення проблем із безпекою, таких як неможливість входу в комп'ютер або

втрата доступу до системи з мережі. У зв'язку з цим, обов'язковою вимогою є проведення тестування всіх змін параметрів безпеки у випадку виконання імпорту та застосування шаблонів безпеки.

Розуміння групової політики

Об'єкти групових політик (GPO) зберігаються одночасно у файловій системі та в базі даних Active Directory. Кожен домен у лісі Active Directory містить повну копію всіх GPO цього домена. Усередині Active Directory зв'язки та відомості про версії GPO зберігаються в розділі бази даних, що містить контекст іменування доменів. Оскільки цей розділ реплікується лише в межах одного домену, завантаження та обробка GPO, що мають зв'язки з іншими доменами (за допомогою сайтів або просто міждоменних зв'язків GPO), може зайняти більше часу [25].

Параметри GPO зберігаються у файловій системі всіх контролерів домену, в папці SYSVOL. Ця папка спільно використовується всіма контролерами домену. У кожного GPO домену є відповідна йому папка, яка знаходиться в підпапці sysvol\companyabc.com\Policies (рис. 13.9).

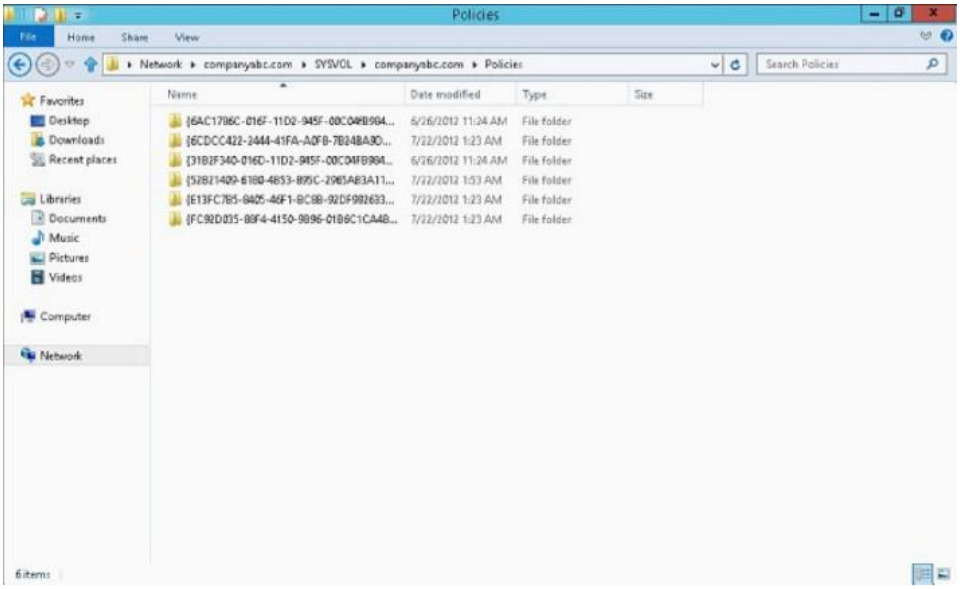

Рисунок 13.9 – Вміст підпапки Policies папки SYSVOL [15]

В якості імені папки GPO використовується глобально унікальний ідентифікатор (Globally Unique Identifier – GUID), привласнений цьому GPO під час його створення. Цей GUID відображається при перегляді властивостей GPO домену в консолі управління груповими політиками [15].

У папці GPO знаходиться звичайний набір підпапок і файлів: папка User, папка Machine (іноді папки ADM, Preferences, Scripts тощо) і файл gpt.ini. Кожна підпапка в ієрархії папок GPO містить файли та папки, пов'язані з конкретним розділом політики або вподобання.

Оскільки об'єкти GPO зберігаються в базі даних Active Directory та у файловій системі контролера домену, вся інформація GPO реплікується контролерами домену. Частина об'єктів GPO домену, що зберігається у файловій системі, реплікується в групі Domain System Volume Distributed File System Replication за допомогою служби реплікації розподіленої файлової системи (Distributed File System Replication – DFSR) [15].

Графік реплікації SYSVOL управляється графіком DFSR, який за замовчуванням збігається з циклом реплікації бази даних Active Directory. Реплікація виконується кожні 5 хвилин або негайно між контролерами домену одного сайту Active Directory та за графіком реплікації зв'язків сайтів між контролерами домену для різних сайтів. Для застарілих доменів замість DFSR використовується служба реплікації файлів (FRS) [15].

Підпапка User (Користувач) містить файли та папки, що використовуються для зберігання параметрів, програм, сценаріїв та інших налаштувань для політик користувача та

об’єкта користувача, сконфігурованих у даному GPO.

Підпапка Machine (Комп’ютер) містить файли та папки, що використовуються для зберігання параметрів, програм, сценаріїв та інших налаштувань для політик комп’ютера або об’єкта комп’ютера, сконфігурованих у даному GPO.

Підпапка Preference (Вподобання) містить файли та папки, що використовуються для зберігання параметрів, завдань і будь-яких інших налаштувань політик для конкретного комп’ютера або об’єкта комп’ютера, сконфігурованих у даному GPO.

Підпапка ADM створюється для нових GPO, якщо в них імпортуються файли адміністративних шаблонів старого формату. Усі GPO, створені за допомогою клієнтських програм Windows 2012 Server і Windows XP або Windows 2022 Server і Windows Server 2025, містять підпапку ADM, де зберігаються всі файли адміністративних шаблонів, які імпортовані в цей GPO і на які є посилання в цьому GPO. Нові об’єкти GPO, створені в Windows Server 2012, за замовчуванням не містять цю підпапку.

У кожній груповій політиці параметри розбиті на кілька розділів. Багато параметрів GPO визначають ключі та значення системного реєстру. Стан і значення таких ключів зберігаються у файлах registry.pol в одній із папок User або Machine. Для оптимізації роботи файл registry.pol містить лише параметри, налаштовані в GPO.

При створенні об’єкта GPO для нього створюється папка в папці SYSVOL пов’язаного з ним контролера домену. У корені цієї папки GPO є файл з іменем gpt.ini, який містить номер ревізії GPO. Цей номер використовується при обробці GPO об’єктом комп’ютера або користувача. Під час першої обробки GPO номер ревізії зберігається в системі, а при наступних обробках номер із файлу gpt.ini порівнюється зі значенням, що зберігається в кеш-пам’яті локальної системи. Якщо цей номер не змінився, деякі частини GPO не обробляються. Однак існують і такі частини GPO, які обробляються завжди – наприклад, сценарії [15].

При кожній зміні GPO номер посилання або ревізії збільшується, і хоча файл gpt.ini містить одне число, воно насправді представляє собою окремий номер ревізії для розділу комп’ютера і користувача даного GPO.

Стандартне налаштування на відсутність обробки деяких розділів GPO при незмінному номері ревізії можна скасувати. У деяких випадках, навіть якщо GPO не змінився, користувач або програма можуть змінити важливі параметри, а іноді просто потрібна примусова обробка всього GPO.

У більшості випадків адміністративні шаблони GPO являють собою набір текстових або XML-файлів, що містять чітко визначені параметри, яким можна присвоїти ряд різних значень. Адміністративні шаблони надають адміністраторам легкий доступ до багатьох конфігураційних параметрів, що зазвичай використовуються для управління комп’ютерами серверів і робочих станцій, а також кінцевими користувачами [15].

При створенні нового GPO в цю політику імпортується базовий набір адміністративних шаблонів або посилання на них. Можна імпортувати й додаткові адміністративні шаблони, щоб додати в будь-яку політику потрібні функції. Коли в існуючій мережі встановлюються нові операційні системи, адміністратори групових політик побачать інші значення в редакторах групових політик при редагуванні політики в новішій ОС. Це може призвести до плутанини та проблем, тому всім адміністраторам слід застосовувати нові адміністративні шаблони. Швидкий спосіб для ефективного використання таких шаблонів в організації – задіяння центрального сховища групових політик і оновлення адміністративних шаблонів у цьому сховищі при появі кожної нової операційної системи.

Як було зазначено раніше, кожен GPO в лісі Active Directory повинен був мати відповідну папку в папці SYSVOL кожного контролера того домену, в якому створювався цей GPO. Якщо контролери конкретного домену працюють під управлінням Windows Server, то кожна з таких папок GPO повинна була містити (в папці ADM) копії всіх адміністративних шаблонів, завантажених у даний GPO. Це призводило до великого обсягу дублювання файлів адміністративних шаблонів, вимагало додаткового місця та збільшувало обсяг реплікації між контролерами доменів.

У новій інфраструктурі групових політик, яка з’явилася в Windows Vista та Windows

Server 2008 і присутня в Windows Server 2025, створені GPO зберігають лише файли та папки, потрібні для зберігання встановлених параметрів, сценаріїв, registry.pol та інших файлів, що мають відношення до GPO. При відкритті GPO для редагування або обробки на комп'ютері Windows Vista, Windows Server 2008 або новішої версії використовується посилання на локальну копію адміністративних шаблонів, але вони не копіюються в папку нового GPO в папці SYSVOL. Натомість у файлах, які зберігаються на локальних робочих станціях або в центральному сховищі домену, є посилання на адміністративні шаблони [13].

Центральне сховище GPO – це файлове сховище, в якому зберігаються всі адміністративні шаблони наступного покоління. Це центральне сховище призначене для зберігання всіх нових адміністративних шаблонів ADMX і ADML, і в кожній робочій станції зберігаються посилання на файли в контролері домену, який застосовується для обробки її групових політик. Тепер при відкритті або обробці GPO система спочатку перевіряє наявність центрального сховища, а потім використовує шаблони, що зберігаються тільки в цьому сховищі. Центральне сховище GPO можна створювати в інфраструктурах Active Directory, де працюють контролери доменів Windows Server 2003 або новішої версії [15].

Консолі управління груповими політиками в Windows Server надають інструмент управління GPO під назвою «стартові об'єкти GPO». Стартові GPO схожі на звичайні GPO, але містять лише параметри, доступні з адміністративних шаблонів. Як і шаблони безпеки, які можна використовувати для імпорту та експорту сконфігурованих параметрів у розділі безпеки політики, стартові GPO можна застосовувати для первинного заповнення розділів Administrative Templates (Адміністративні шаблони) вузлів GPO під назвами Computer Configuration (Конфігурація комп'ютера) і User Configuration (Конфігурація користувача) [15].

Після появи Windows Server 2008 компанія Microsoft випустила набір визначених стартових GPO, які зараз входять до складу Windows Server 2025 і які можна отримати на основі інформації з керівництва Microsoft з безпеки клієнтів у Windows XP та Windows. Ці стартові GPO є політиками лише для читання, але адміністратори мають можливість створювати власні стартові GPO, необхідні в організації. Активація функціональності стартових GPO, створення та управління ними описані в розділі «Створення та застосування стартових об'єктів GPO» далі.

Параметри політики – це параметри, доступні для налаштування в конкретному об'єкті GPO. Ці параметри беруться з базових адміністративних шаблонів, налаштувань безпеки, сценаріїв, якості обслуговування (QoS), заснованого на політиці, і, в деяких випадках, з пакетів розгортання ПЗ. Багато параметрів політик відповідають «один до одного» деяким ключам і значенням системного реєстру. Для різних параметрів існують різні допустимі значення, у тому числі й довільний текст.

Параметри політик GPO зазвичай мають одне з трьох значень: не вказано, увімкнено або вимкнено. Адміністраторам дуже важливо не тільки усвідомити різницю між цими трьома значеннями, а й знати, чим керує кожен параметр політики. Наприклад, параметр політики, що забороняє доступ до панелі управління, блокує доступ, коли він увімкнений, і дозволяє доступ, якщо він вимкнений.

Параметри політик GPO застосовуються до об'єктів комп'ютерів або користувачів. Адміністратор може виявити в одному й тому ж GPO певний параметр політики і у вузлі конфігурації комп'ютера, і у вузлі конфігурації користувача. У таких випадках, тобто коли параметр політики заданий для обох об'єктів, і політика пов'язана з об'єктом користувача та робочої станції, в яку входить користувач, налаштування комп'ютера перекриває налаштування користувача.

У групових політик є два основних вузли налаштувань – Computer Configuration (Конфігурація комп'ютера) та User Configuration (Конфігурація користувача). Кожен із них містить два інші вузли – Policies (Політики) та Preferences (Вподобання/Рекомендовані).

Вузол Policies (Політики), який знаходиться в конфігурації групових політик як для комп'ютерів, так і для користувачів, містить параметри, які здебільшого застосовуються примусово і не можуть переналаштовуватися клієнтами. Якщо параметри можуть мати кілька значень, то параметри з вузлом Policies обов'язково застосовуються до клієнта, але

адміністратори можуть додати або змінити і свою частину налаштувань. Наприклад, якщо присвоєння прав користувачеві виконано за допомогою доменної політики, адміністратор не зможе видалити елементи цих прав, призначені з політики, але він зможе додавати та змінювати інші елементи. Вузол Policies містить налаштування безпеки (брандмауера та мережі), але основний обсяг налаштувань міститься в розділі Administrative Templates (Адміністративні шаблони) [15].

Вузол Preferences (Вподобання), який знаходиться в конфігурації групових політик як для комп'ютерів, так і для користувачів, містить параметри, які здебільшого раніше не були присутні в групових політиках і управляються спеціальними сценаріями та адміністративними шаблонами. Параметри вподобань задаються початково, але зазвичай кінцевий користувач може змінити їх після обробки групових політик [15].

Тема 14
Служба DNS, DHCP

Поняття системи доменних імен DNS

Система доменних імен (DNS) виникла з проєкту ARPANET у 1960-х роках, вирішуючи потребу у зручному для користувача способі ідентифікації мережевих пристроїв, що виходить за межі числових IP-адрес. Ця концепція еволюціонувала в DNS у тому вигляді, в якому вона відома сьогодні, на початку 1980-х років із випуском фундаментальних специфікацій, задокументованих у запитах на коментарі (RFC). DNS організована в ієрархічну структуру, подібну до дерева, де коренева зона розгалужується на різні домени та піддомени, кожен з яких містить записи ресурсів, що надають важливу інформацію про мережеві ресурси [13].

Доменне ім'я конструюється з декількох сегментів, відомих як мітки, що розділені крапками – наприклад, packtpub.com. Ця система спирається на розподілену базу даних, що використовує архітектуру клієнт-сервер, де мережеві хости виступають у ролі серверів імен. Ці сервери відповідають за перетворення (розв'язання) доменних імен у відповідні IP-адреси, забезпечуючи безперебійну навігацію та з'єднання в інтернеті. Такий ієрархічний і розподілений підхід підвищує масштабованість, ефективність та надійність управління доменними іменами та мережевими ресурсами [13].

Для повного розуміння того, як функціонує DNS, доцільно простежити послідовність кроків, що відбуваються під час спроби доступу до вебсайту. DNS є необхідною для трансляції зручних для людини доменних імен у машинозчитувані IP-адреси, полегшуючи комунікацію між користувачами та вебсайтами. Процес розв'язання імен DNS, що описує механізм пошуку браузером коректної IP-адреси для з'єднання при введенні веб-адреси, такої як www.packtpub.com, складається з кількох етапів.

Процедура ініціюється введенням URL-адреси: коли адреса www.packtpub.com вводиться в адресний рядок браузера і натискається клавіша Enter, браузер надсилає запит на з'єднання з цим доменом. Цей запит спершу надходить до критично важливого компонента інфраструктури DNS, відомого як рекурсивний перетворювач (резолвер). Зазвичай керований інтернет-провайдером (ISP), цей резолвер відповідає за обробку запитів від імені клієнта. Далі рекурсивний резолвер комунікує з глобальними кореневими серверами, які зберігають інформацію про домени верхнього рівня (TLD), такі як .com. Ці сервери не володіють повною інформацією DNS, проте вони спрямовують резолвер до відповідних серверів TLD.

Сервери TLD, у свою чергу, відповідають наданням інформації, що спрямовує резолвер до авторитетних серверів імен для конкретного домену, наприклад, packtpub.com. Після цього резолвер опитує ці авторитетні сервери імен для знаходження точної IP-адреси, асоційованої з packtpub.com. Авторитетні сервери містять фактичні записи DNS, що зіставляють доменні імена з IP-адресами. Як тільки резолвер отримує IP-адресу вебсервера, що хостить packtpub.com, він передає цю інформацію назад у браузер. Маючи IP-адресу, браузер може встановити з'єднання з вебсервером і отримати контент вебсайту для перегляду [13].

Цей поетапний процес ілюструє складні механізми роботи DNS, підкреслюючи її роль у перетворенні доменних імен в IP-адреси, що уможливлюють безперебійну комунікацію в інтернеті. Розуміння цього процесу підкреслює важливість правильного налаштування ролі DNS у мережі. Виконуючи це, забезпечується ефективне розв'язання доменних імен, що є критично важливим як для операцій внутрішньої мережі, так і для зовнішнього доступу до інтернету.

Структура простору імен

Структура системи доменних імен (DNS) нерозривно пов'язана з архітектурою мережі Інтернет, через що ці поняття часто ототожнюються. Така структура зарекомендувала себе як надзвичайно зручна, а той факт, що вона продовжує користуватися популярністю протягом тривалого часу, лише підтверджує її функціональність та надійність. Для формування більш широкого уявлення про те, яким чином служба DNS інтегрується в середовище Windows Server, необхідне детальне вивчення компонентів, з яких складається DNS, а також засобів їх

об’єднання в єдину логічну структуру.

Обмежена область, що визначається іменем DNS, класифікується як простір імен DNS. Прикладами таких просторів можуть слугувати домени microsoft.com або marketing.companyabc.com. Простори імен поділяються на загальнодоступні (публічні) та внутрішні. Загальнодоступні простори імен публікуються в мережі Інтернет і функціонування їх залежить від дотримання низки стандартів. Усі простори імен доменів верхнього рівня, таких як .com, .net, .org тощо, є зовнішніми або загальнодоступними. Стосовно внутрішніх просторів імен, слід зазначити, що вони не публікуються в Інтернеті, а отже, не обмежуються жорсткими правилами реєстрації. Тобто внутрішній, не опублікований простір імен може мати будь-який вигляд, наприклад dnsname.local або companyabc.internal. Найчастіше внутрішні простори імен використовуються в службах Active Directory, оскільки це підвищує рівень безпеки середовища. Враховуючи, що такі простори імен не публікуються, пряме звернення до них із мережі Інтернет є неможливим [13].

У контексті концепції просторів імен DNS, простір імен являє собою обмежену логічну область, що утворюється іменем DNS та його піддоменами. Наприклад, імена europe.companyabc.com, asia.companyabc.com та companyabc.com розглядаються як частини одного й того ж безперервного простору імен DNS. Простір імен DNS у доменних службах Active Directory (AD DS) може бути опублікований в Інтернеті (на кшталт microsoft.com або msn.com) або ж прихований від зовнішнього доступу, що залежить від обраної стратегії та вимог безпеки, які реалізуються адміністраторами системи.

Зовнішні (опубліковані) простори імен визначаються як імена DNS, що розпізнаються з будь-якої точки мережі Інтернет. Подібні простори імен раніше часто застосовувалися в організаціях, які з метою уніфікації прагнули, щоб їхнє доменне ім’я, яке зазвичай використовується в Інтернеті, представляло також і структуру AD DS. Однак практика демонструє, що така модель є недостатньо зручною та безпечною. Оскільки питання безпеки відіграють дедалі важливішу роль, систему DNS рекомендується встановлювати як окремий компонент: наявність внутрішніх зон AD DNS із можливістю доступу до них із мережі Інтернет не рекомендується [13].

Внутрішні (приховані) простори імен використовуються організаціями, для яких публікація внутрішньої доменної структури є неприпустимою з точки зору інформаційної безпеки. Такі організації можуть визначати схеми AD DS із внутрішнім простором імен, який не доступний для читання із зовнішньої мережі. Наприклад, компанія може володіти зовнішнім простором імен DNS cco.com, тоді як структура AD DS відповідатиме простору імен cco.internal або іншому подібному. Для внутрішніх просторів імен допускається будь-яка комбінація, адже в них відсутні обмеження на використання доменів .com, .net, .gov тощо. За потреби домен можна назвати навіть довільним чином, наприклад ilovemydomain.verymuch (хоча такий підхід не є рекомендованим). З практичних міркувань для приватної адресації спеціально зарезервовано простір імен .internal, використання якого у багатьох випадках є доцільним та зручним [13].

Слід звернути увагу на важливий аспект уникнення конфліктів імен. Якщо приймається рішення використовувати простір доменних імен, який теоретично може бути – на даний момент або в майбутньому – придбаний та застосований в Інтернеті, то для уникнення можливих колізій при перетворенні (розворот) імен рекомендується одразу набути права на це доменне ім’я. Наприклад, при виборі в якості внутрішнього простору імен companyabc.com доцільно перевірити його доступність і, за можливості, зареєструвати його. Якщо виявиться, що цим ім’ям домену вже володіє інша організація, рекомендується обрати для простору імен своєї AD DS інше доменне ім’я. Навіть якщо домен не публікується в Інтернеті, у користувачів домашніх або портативних комп’ютерів, яким потрібен доступ до домену через комутоване з’єднання або мережу VPN, можуть виникати конфлікти через помилкове спрямування запитів у простір імен DNS в Інтернеті, а не в локальний простір імен компанії.

Встановлення та налаштування служби DNS у Windows Server

В операційній системі Windows Server 2025 роль DNS є важливою для забезпечення можливості трансляції доменних імен в IP-адреси, що сприяє безперебійній мережевій

комунікації та доступу до ресурсів. Конфігурація цієї ролі може бути здійснена за допомогою інструменту Server Manager («Диспетчер серверів»). Як зображено на рисунку 14.1, процес розпочинається з доступу до Server Manager та вибору опції додавання ролей і компонентів.

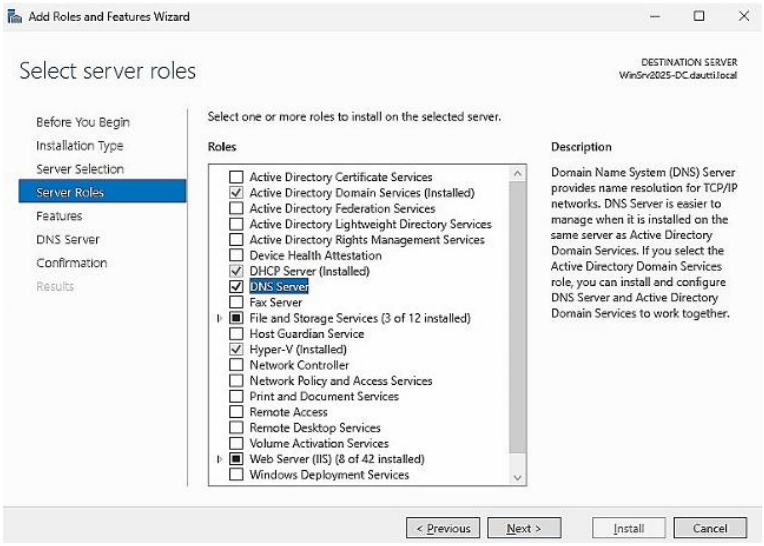

Рисунок 14.1 – Встановлення ролі DNS [13]

Встановлення ролі DNS може бути виконане як у вигляді незалежної служби, так і у поєднанні з доменними службами Active Directory (AD DS). У випадку окремого встановлення роль DNS функціонує автономно для обробки запитів на розв'язання доменних імен. Однак інтеграція DNS з AD DS, значно розширює загальну функціональність мережі, дозволяючи серверу DNS підтримувати операції Active Directory, такі як визначення розташування контролерів домену (DC) та пошук записів служб (рис. 14.2). Така інтеграція є особливо корисною для управління масштабними мережами, де AD та DNS взаємодіють для оптимізації операцій [29].

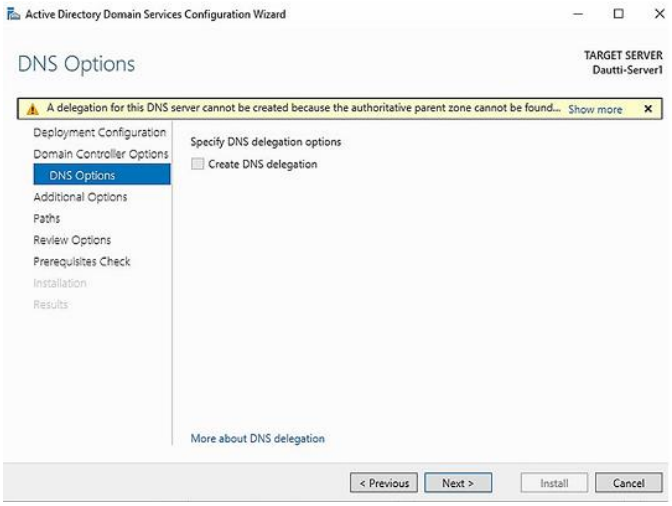

Рисунок 14.2 – Додавання DNS разом з AD DS [13]

Більше того, роль DNS часто включається як складова частина процесу встановлення AD DS, забезпечуючи цілісне налаштування, що підтримує розв'язання доменних імен у середовищі Active Directory. Такий підхід гарантує належну конфігурацію служб DNS для забезпечення потреб AD, включаючи автоматичне створення необхідних DNS-записів. Після успішного встановлення та налаштування ролі DNS з'являється можливість керувати розв'язанням доменних імен та підвищувати функціональність мережі.

Після інсталяції виконується подальша конфігурація сервера. Насамперед, здійснюється налаштування мережевих інтерфейсів. За замовчуванням сервер DNS прослуховує запити на всіх інтерфейсах IP-адрес. Існує можливість налаштувати сервер DNS

на прослуховування лише визначеного інтерфейсу, використовуючи графічний інтерфейс користувача (GUI) або засоби PowerShell [30].

Для налаштування інтерфейсу, що використовується для прослуховування DNS-запитів через консоль DNS Manager, необхідно виконати наступні дії: у меню «Пуск» (Start) обирається пункт «Засоби адміністрування Windows» (Windows Administrative Tools), а потім – DNS. Далі обирається потрібний сервер, на якому (через утримання натискання або клік правою кнопкою миші) викликається контекстне меню та обирається пункт «Властивості» (Properties). Щоб обмежити сервер DNS використанням конкретної IP-адреси, вибирається опція «Лише вказані IP-адреси» (Only the following IP address), зазначається необхідна адреса, після чого натискається кнопка OK [13].

Наступним етапом є конфігурація кореневих посилань. Сервери кореневих посилань використовуються для допомоги у розв'язанні адресної інформації DNS у випадках, коли сервер DNS не в змозі розв'язати запит локально за допомогою розміщеної зони або кешу сервера. Сервери імен кореневих посилань заповнюються за замовчуванням під час нових інсталяцій. За необхідності список кореневих серверів імен можна редагувати, перейшовши на вкладку «Кореневі посилання» у діалоговому вікні властивостей сервера DNS або використовуючи PowerShell.

Важливо зазначити, що видалення всіх серверів кореневих посилань не підтримується. Замість цього сервер DNS налаштовується на відмову від використання серверів кореневих посилань шляхом вибору опції «Вимкнути рекурсію» (Disable recursion) на вкладці «Додатково» (Advanced) консолі DNS Manager. Вимкнення рекурсії також деактивує будь-які налаштовані сервери пересилання. Альтернативно, можна зняти позначку з пункту «Використовувати кореневі посилання, якщо сервери пересилання недоступні» (Use root hints if no forwarders are available) на вкладці «Сервери пересилання» (Forwarders).

Процедура редагування кореневих посилань через консоль DNS Manager виглядає наступним чином: відкривається консоль DNS, у властивостях сервера обирається вкладка «Кореневі посилання» (Root Hints), виділяється елемент для редагування та натискається кнопка «Змінити» (Edit). Далі вводиться повне доменне ім'я (FQDN) та натискається «Розв'язати» (Resolve). Після перевірки та, за необхідності, редагування IP-адреси натискається OK. Після перегляду оновленого списку серверів кореневих посилань підтверджується завершення операції натисканням OK. Варто звернути увагу, що ім'я сервера має завершуватися крапкою.

Також передбачена можливість налаштування серверів пересилання. Сервер пересилання конфігурується опціонально для розв'язання адресної інформації DNS замість пересилання трафіку до кореневих серверів DNS. Додавання серверів пересилання здійснюється через GUI або за допомогою командлета PowerShell Set-DNSServerForwarder. Слід зауважити, що кореневі посилання DNS не використовуються, доки сервери пересилання відповідають на запити. Для налаштування серверів пересилання через консоль DNS Manager у властивостях сервера обирається вкладка «Сервери пересилання» (Forwarders) та натискається кнопка «Змінити» (Edit). Вводиться IP-адреса DNS-сервера, до якого будуть пересилатися запити. Цей крок повторюється необхідну кількість разів. Після натискання OK та перегляду списку серверів DNS конфігурація завершується натисканням кнопок OK або «Застосувати» (Apply).

Створення зон і записів

У контексті адміністрування служби DNS ключовим етапом є створення зон, які виступають логічними контейнерами для зберігання записів ресурсів. Основна зона (Primary Zone) містить головну копію даних, доступну для читання та запису. У середовищі Windows Server існує два механізми зберігання даних основної зони: інтеграція з Active Directory (AD) та файлове зберігання.

Інтеграція зон DNS із Active Directory забезпечує низку переваг, зокрема використання механізмів мульти-майстерної реплікації, посилену безпеку та спрощене адміністрування. Для створення такої зони використовується консоль управління DNS (DNS Manager) або засоби автоматизації PowerShell. Процедура створення через графічний інтерфейс передбачає виконання наступних кроків.

У консолі DNS Manager здійснюється підключення до цільового сервера, після чого у контекстному меню обирається пункт «New Zone» (Нова зона), що ініціює запуск «Майстра створення нових зон» (New Zone Wizard) (рис. 14.3). На етапі вибору типу зони (Zone Type) необхідно обрати «Primary zone» (Основна зона) та переконатися, що активовано опцію «Store the zone in Active Directory» (Зберігати зону в Active Directory). Слід зазначити, що дана опція доступна виключно у випадку, коли сервер DNS функціонує як контролер домену AD DS [13].

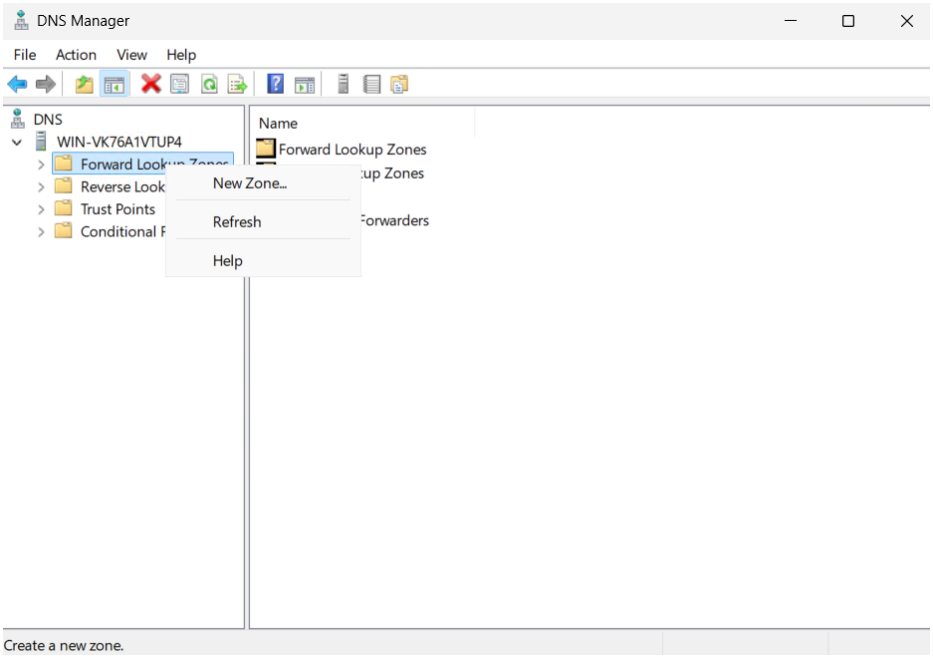

Рисунок 14.3 – Початок створення прямої зони DNS [13]

Критично важливим етапом є визначення області реплікації зони (Active Directory Zone Replication Scope). Адміністратору пропонуються наступні варіанти розповсюдження даних:

– усі сервери DNS, що працюють на контролерах домену в даному лісі;
– усі сервери DNS, що працюють на контролерах домену в даному домені;
– усі контролери домену в даному домені (опція для сумісності з Windows 2000);
– усі контролери домену, включені до специфічного розділу каталогу.

Далі визначається тип перегляду: зона прямого перегляду (Forward lookup zone) для перетворення імен в IP-адреси або зона зворотного перегляду. Після введення імені зони (наприклад, north.contoso.com) налаштовується режим динамічних оновлень (Dynamic Update). Для інтегрованих зон Active Directory рекомендується вибір опції «Allow only secure dynamic updates» (Дозволяти тільки безпечні динамічні оновлення), що гарантує автентифікацію комп'ютерів перед внесенням змін до записів. Інші варіанти включають дозвіл будь-яких оновлень (що знижує рівень безпеки) або повну заборону динамічних оновлень [30].

У сценаріях, де інтеграція з AD неможлива або недоцільна, створюється стандартна основна зона, дані якої зберігаються у текстовому файлі. Процес створення аналогічний попередньому, проте на етапі вибору типу зони опція «Store the zone in Active Directory» має бути деактивованою (знятою).

При конфігуруванні такої зони системі необхідно вказати ім'я файлу зони. За замовчуванням пропонується створити новий файл з іменем зони та розширенням .dns (наприклад, east.contoso.com.dns), який буде розміщено у системній директорії %SystemRoot%\system32\dns. Також існує можливість імпорту існуючого файлу зони, попередньо скопійованого у зазначену директорію. Важливою відмінністю від інтегрованих зон є налаштування динамічних оновлень: опція безпечних оновлень (Secure dynamic updates) у даному випадку недоступна, тому вибір обмежується дозволом небезпечних і безпечних оновлень або їх забороною. Завершується процедура натисканням кнопки «Finish» [13].

Вторинна зона (Secondary Zone) являє собою копію основної зони, доступну лише для читання. Вона використовується для розподілу навантаження, забезпечення відмовостійкості та зменшення трафіку запитів у глобальній мережі [13].

Для створення вторинної зони у «Майстрі створення нових зон» обирається тип «Secondary zone». На етапі іменування необхідно вказати ім'я, яке точно відповідає імені основної зони, з якої буде здійснюватися реплікація (наприклад, south.contoso.com). Ключовим налаштуванням є визначення головних DNS-серверів (Master DNS Servers) [13].

У відповідному діалоговому вікні вводяться IP-адреси одного або декількох серверів, що містять копію основної зони (рис. 14.4). Необхідною умовою успішного розгортання вторинної зони є налаштування дозволу на передачу зон (Zone Transfer) на стороні основного сервера для IP-адреси вторинного сервера. Після валідації введених адрес процес завершується натисканням кнопки «Finish».

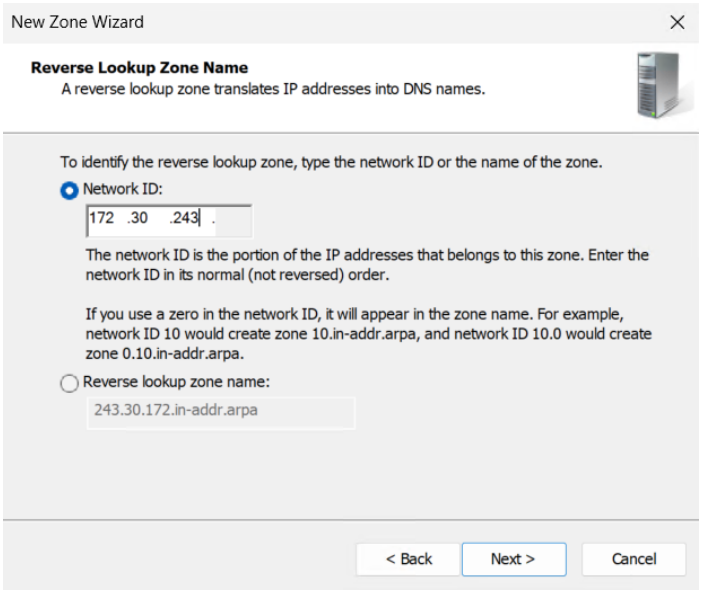

Рисунок 14.4 – Надання ідентифікатора мережі для вторинної зони DNS [13]

Делегування зон є фундаментальним механізмом ієрархічної структури DNS, що дозволяє передати відповідальність за частину простору імен (піддомен) іншому DNS-серверу. Це забезпечує децентралізацію управління та підвищення ефективності розв'язання імен.

Процедура делегування виконується через консоль DNS Manager шляхом вибору батьківської зони та активації пункту «New Delegation» (Нове делегування) у контекстному меню. У діалоговому вікні «Delegated Domain Name» вводиться ім'я делегованого піддомену (наприклад, для south.west.contoso.com вводиться south), при цьому повне доменне ім'я (FQDN) формується автоматично.

Наступним кроком є визначення серверів імен, які будуть авторитетними для делегованої зони. Необхідно натиснути кнопку «Add» та вказати FQDN або IP-адресу відповідного DNS-сервера. Система автоматично намагається розв'язати введене ім'я в IP-адресу (Resolve). Після успішної валідації та додавання необхідної кількості серверів натискається кнопка «OK» та «Finish» для завершення роботи майстра. У результаті в батьківській зоні створюються записи NS (Name Server), які вказують на сервери, що обслуговують делегований піддомен, а також, за необхідності, записи прив'язки (Glue records) [13].

В основі функціонування системи доменних імен лежать записи ресурсів (Resource Records – RR), які слугують фундаментальними ідентифікаторами об'єктів у мережі. Кожен такий запис є унікальним у межах свого домену і використовується для виконання пошукових запитів, пов'язуючи зрозумілі імена з технічними ресурсами. Враховуючи розподілену природу та ієрархічність DNS, ідентичні записи можуть існувати на різних рівнях структури, проте в конкретній зоні вони виконують чітко визначену роль. У більшості

сучасних реалізацій, зокрема інтегрованих зі службами Active Directory в Windows Server, адміністраторам доводиться працювати зі спеціалізованим набором записів, розуміння яких є критичним для забезпечення стабільності мережевої інфраструктури.

Ключовим елементом будь-якої зони є запис початку повноважень (Start of Authority – SOA), який визначає сервер, що виступає первинним джерелом інформації та відповідає за оновлення даних у цій зоні. Цей запис містить життєво важливі параметри: час життя (TTL) для кешування, контактні дані відповідального адміністратора та серійний номер зони. У середовищі Windows Server запис SOA створюється автоматично під час ініціалізації ролі DNS для Active Directory, заповнюючись стандартними значеннями, які згодом можна адаптувати до політик організації через консоль управління [13].

Найбільш масовим типом записів, що становить основу адресації в інтернеті, є записи хостів типу A, які прямо зіставляють доменне ім’я з IPv4-адресою пристрою (рис. 14.5). Саме ці записи дозволяють користувачам знаходити ресурси за іменами. Варто зазначити, що окрім базового зіставлення, записи можуть містити додаткові метадані, такі як точний час створення або індивідуальні налаштування TTL. Однак, щоб переглянути або змінити ці розширені параметри в консолі DNS Management, адміністратору необхідно увімкнути режим розширеного перегляду (View – Advanced), оскільки в стандартному режимі відображається лише базова інформація [15].

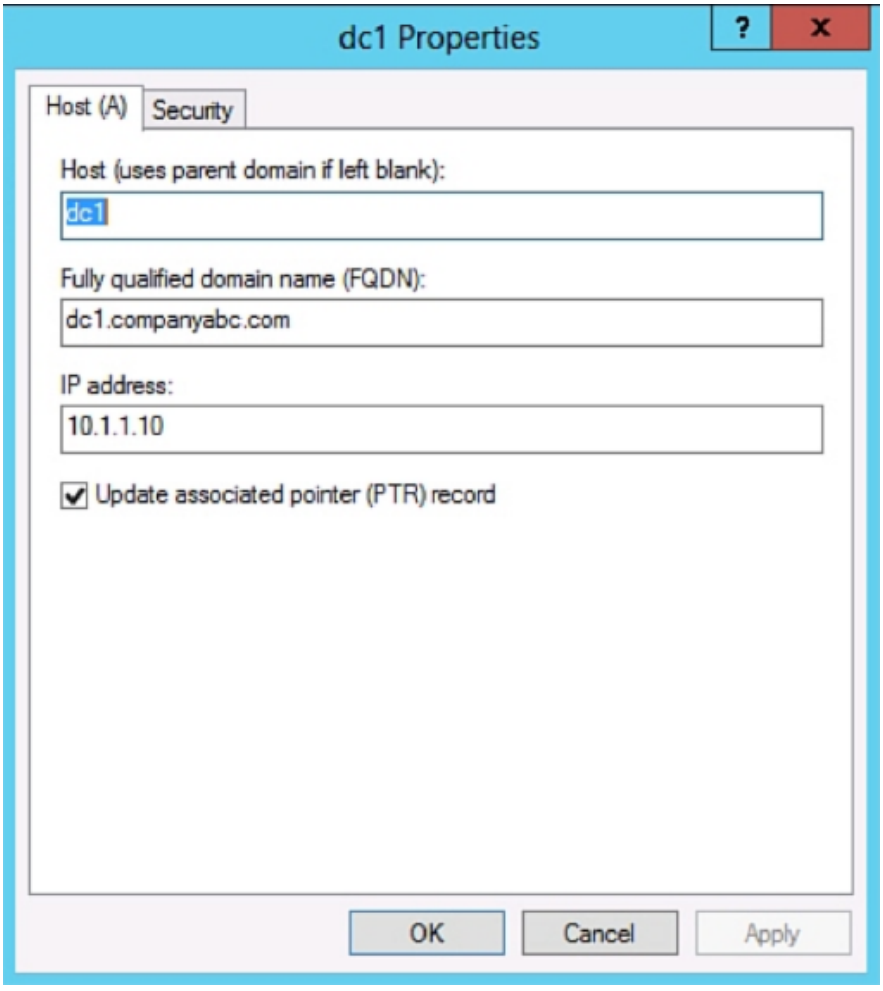

Рисунок 14.5 – Записи хостів типу A [15]

Для забезпечення доступності зони використовуються записи серверів імен (Name Server – NS), які вказують на те, які саме сервери уповноважені обробляти запити для даного домену. На відміну від запису SOA, який є єдиним для зони, записів NS зазвичай створюють декілька для забезпечення відмовостійкості. Важливо розуміти технічний нюанс: запис NS вказує не на IP-адресу, а на доменне ім’я сервера, тому для коректної роботи механізму розв’язання імен необхідна наявність відповідного A-запису для кожного сервера імен, на який посилається запис NS.

Особливе значення для інфраструктури Microsoft Active Directory мають записи служб (Service – SRV), які дозволяють клієнтам локалізувати сервери, що надають специфічні сервіси, такі як LDAP, Kerberos або глобальний каталог. Цей тип запису містить детальну технічну інформацію, включаючи порт, пріоритет та вагу служби, що дозволяє ефективно розподіляти навантаження між контролерами домену (рис. 14.6). Оскільки підтримка SRV-записів з'явилася в стандартах DNS не відразу, при використанні сторонніх DNS-серверів (наприклад, на базі UNIX BIND) критично важливо переконатися, що версія програмного забезпечення (8.1.2 або вище) підтримує цей стандарт, інакше коректна робота домену Windows буде неможливою [15].

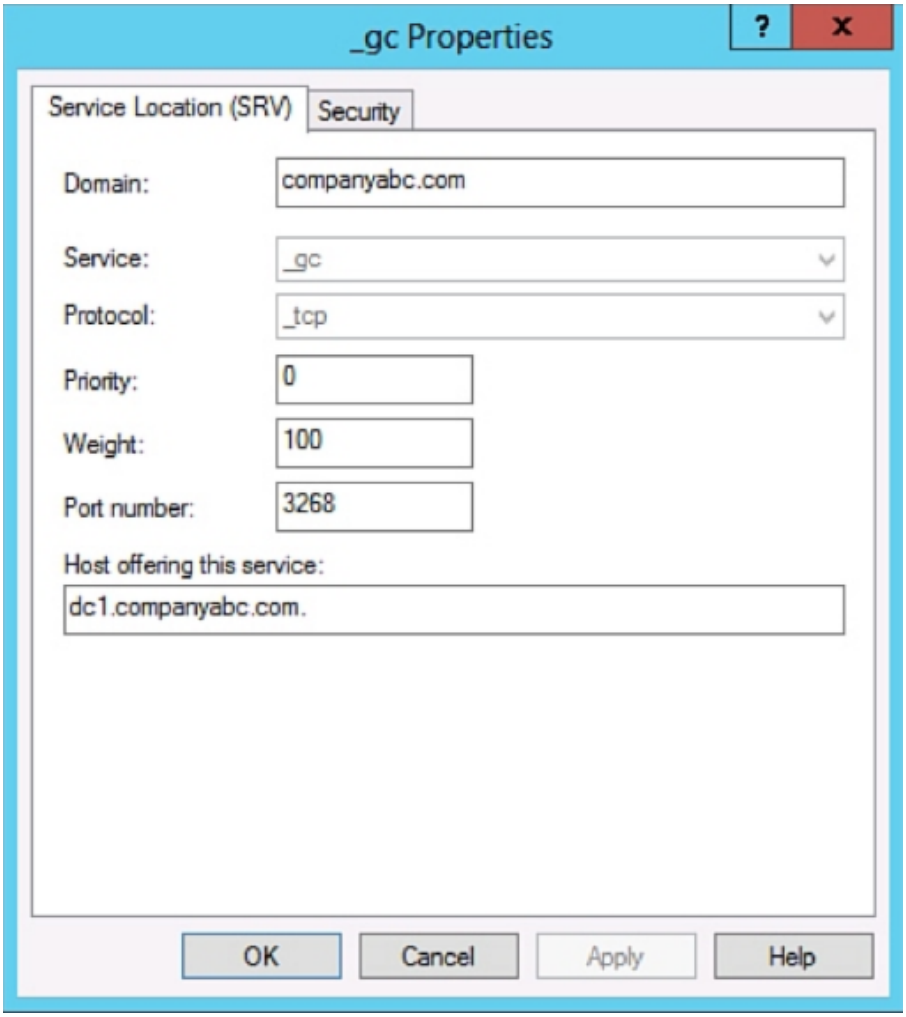

Рисунок 14.6 – Приклад запису SRV для елемента глобального каталога Active Directory [15]

Окрім інфраструктурних записів, існують спеціалізовані типи для маршрутизації пошти та керування псевдонімами. Записи обміну поштою (MX) визначають сервери, що приймають SMTP-трафік для домену, причому пріоритет обробки пошти регулюється числовим значенням у записі. Для створення альтернативних імен вузлів використовуються записи канонічних імен (CNAME), які діють як псевдоніми, перенаправляючи запит на основний A-запис хоста. Це особливо зручно при міграції сервісів або для створення простих імен на кшталт mail.company.com замість складних технічних ідентифікаторів серверів.

Завершують екосистему DNS записи, що виконують специфічні або допоміжні функції. Для зворотного перетворення IP-адрес в імена використовуються записи покажчиків (PTR), які розміщуються виключно в зонах зворотного перегляду і є необхідними для багатьох механізмів безпеки. Також, з огляду на поступовий перехід мереж на новий протокол, зростає значення записів AAAA, які виконують ту ж функцію, що й записи A, але для 128-бітних адрес IPv6. Серед більш рідкісних типів можна виділити записи ISDN для телефонії, KEY для зберігання ключів шифрування та інші спеціалізовані формати, що застосовуються для вузьких задач адміністрування.

Розуміння DNS-зон

Зони становлять основу ієрархічної структури DNS, яка регулює процеси розв'язання доменних імен у мережевому середовищі. Зони DNS є невід'ємною частиною простору імен Active Directory, який тісно узгоджується із глобальним простором імен DNS, забезпечуючи структурований та масштабований підхід до управління даними, пов'язаними з доменами. Шляхом сегментації зон DNS адміністратори отримують можливість більш ефективно зберігати інформацію про конкретні домени та управляти нею, що гарантує точність та ефективність процесу розв'язання доменних імен.

В архітектурі DNS виділяють три основні типи зон, кожен з яких виконує чітко визначену функцію.

Першим типом є первинна зона (Primary zone), яка виступає авторитетним джерелом DNS-інформації для домену. Вона містить остаточну, доступну для редагування копію бази даних DNS і відповідає за підтримку всіх записів ресурсів у межах своєї області дії. Ця зона є центральним органом управління для розв'язання імен у домені, забезпечуючи коректність та послідовність відповідей на DNS-запити [13].

Другим типом є вторинна зона (Secondary zone), яка діє як резервна копія первинної зони та містить копію DNS-записів, призначену лише для читання. Ця зона має критичне значення для забезпечення надлишковості, оскільки вона дозволяє процесу розв'язання імен продовжуватися безперервно навіть у випадку недоступності первинної зони. Вторинна зона синхронізується з первинною, що гарантує відображення в ній найбільш актуальної інформації DNS [13].

Третім спеціалізованим варіантом є зона-заглушка (Stub zone), яка є різновидом вторинної зони. На відміну від стандартної вторинної зони, що містить повну копію бази даних DNS, зона-заглушка зберігає лише мінімально необхідний обсяг інформації – зокрема, IP-адреси авторитетних DNS-серверів для даної зони – для перенаправлення запитів до відповідного авторитетного сервера. Така архітектура робить зони-заглушки корисними для спрощення адміністрування DNS та оптимізації мережевого трафіку шляхом зменшення потреби у повній реплікації даних DNS [13].

Ключову роль в управлінні цими зонами відіграють DNS-сервери. Авторитетний DNS-сервер, який оперує записами DNS для конкретного домену, є критично важливим елементом цієї структури. Конфігурація такого сервера може здійснюватися системним адміністратором вручну, що дозволяє забезпечити точний контроль над записами DNS, або динамічно іншими DNS-серверами за допомогою передачі зон та оновлень. Авторитетний сервер виступає кінцевим арбітром для DNS-запитів у своєму домені, гарантуючи точність та актуальність відповідей. На противагу цьому, неавторитетний DNS-сервер покладається на кешовані дані, отримані в результаті попередніх пошуків DNS, і не зберігає оригінальних записів. Хоча неавторитетні сервери можуть надавати швидкі відповіді на основі кешованої інформації, вони не є остаточним джерелом для розв'язання імен, що іноді може призводити до отримання застарілих або неточних відповідей у разі неналежного обслуговування кешу [30].

Поза межами розгляду зон DNS важливо також розуміти роль WINS – застарілої служби, яка здійснює розв'язання імен NetBIOS. Хоча в сучасних мережах DNS значною мірою витіснила WINS, ця служба залишається актуальною в середовищах, де старіші системи та додатки все ще покладаються на NetBIOS для розв'язання імен. Ознайомлення з принципами роботи WINS може бути особливо важливим у мережах, що підтримують застарілу інфраструктуру, оскільки це забезпечує ефективну комунікацію між усіма системами, як старими, так і новими.

Розуміння DNS-запитів

Оскільки основним призначенням системи доменних імен (DNS) є перетворення імен для запитуючих клієнтів, механізм обробки запитів розглядається як один із найбільш фундаментальних елементів архітектури системи. У переважній більшості випадків до бази даних DNS надходять запити двох основних типів: рекурсивні та ітеративні.

Рекурсивні запити найчастіше ініціюються розпізнавачами – клієнтськими компонентами, які потребують розв'язання конкретного доменного імені сервером DNS. Крім того, такі запити можуть генеруватися самим DNS-сервером у випадках, коли налаштовано

використання ретрансляторів на певний визначений сервер імен. Суть рекурсивного запиту полягає у з'ясуванні можливості конкретного сервера імен виконати повне перетворення для специфічного запису, при цьому відповідь може бути або остаточно позитивною, або негативною [13].

При виконанні ітеративних запитів до DNS-сервера ставиться вимога або здійснити перетворення імені, або надати посилання на інший DNS-сервер, який з високою ймовірністю володіє точнішою інформацією щодо місця обробки даного запиту. Після отримання посилання ініціюється наступний ітеративний запит до зазначеного сервера, і цей процес повторюється циклічно до моменту отримання позитивного або негативного результату розв'язання імені. Ці функції лежать в самій основі функціонування DNS, враховуючи її розподілену природу, і дозволяють ефективно виконувати процеси пошуку.

Розуміння еволюції Microsoft DNS

Реалізація доменних служб Active Directory (AD DS), що пропонується в Windows Server 2025, розширює спектр додаткових компонентів, який був впроваджений ще у версії Windows 2000 Server DNS та згодом удосконалювався у Windows Server 2003 і Windows Server 2008, Windows Server 2012 та Windows Server 2022.

У систему AD DS додано низку важливих функціональних покращень, проте вони не є настільки радикальними, щоб суттєво змінити стратегічні рішення щодо архітектури DNS. У цьому контексті розглядаються можливості, які були перенесені у Windows Server 2025 DNS із попередніх версій платформи, і які допомагають відрізняти дану реалізацію від інших систем DNS [13].

Найбільш вагомою зміною в реалізації DNS у Windows 2000 Server стало впровадження концепції зон, вбудованих у каталог, які отримали назву інтегрованих в Active Directory зон. Такі зони зберігалися безпосередньо в базі даних Active Directory, а не у стандартних текстових файлах, як у традиційній DNS. Під час реплікації Active Directory здійснювалася також і реплікація зони DNS. Такий підхід уможливив виконання безпечних оновлень, використання протоколу аутентифікації Kerberos та реалізацію мультимайстерної моделі DNS, за якої жоден із серверів не виступає єдиним еталоном, а на кожному вузлі міститься доступна для запису копія зони. У версіях Windows Server 2012 та пізніших, як і в Windows Server 2008, інтегровані в AD зони DNS продовжують використовуватися, проте з однією суттєвою модифікацією: для зниження навантаження, пов'язаного з реплікацією, інформація таких зон тепер зберігається не в контекстах іменування Active Directory, а у розділі додатків [15].

Механізм динамічних оновлень, що реалізується за допомогою Dynamic DNS (DDNS), надає клієнтам можливість автоматично реєструвати, оновлювати та скасовувати реєстрацію записів хостів під час підключення до мережі. Ця концепція, вперше представлена у Windows 2000 Server DNS, була перенесена у Windows Server 2025 без змін.

Окрім того, введена у Windows 2000 Server та збережена у Windows Server 2025 підтримка розширених наборів символів Unicode дозволяє службі DNS зберігати записи, що складаються із символів Unicode, фактично охоплюючи декілька наборів символів із безлічі різних мов. Це дозволяє DNS-серверу оперувати записами, що містять нестандартні символи, такі як підкреслення, літери різних алфавітів тощо [13].

Незважаючи на наявну підтримку символів Unicode в реалізації Microsoft DNS, у будь-якій інфраструктурі DNS рекомендується застосовувати стандартний набір символів. Дотримання цієї рекомендації забезпечує можливість обміну даними зон із реалізаціями DNS, що не підтримують Unicode, наприклад, із серверами Unix BIND. До стандартного набору входять символи латинського алфавіту (a-z, A-Z), цифри (0-9) та дефіс (-).

Призначення та принцип роботи DHCP

У складі операційної системи Windows Server служба DHCP (Dynamic Host Configuration Protocol) реалізована як опціональна мережева серверна роль, яка розгортається для управління розподілом IP-адрес та надання іншої інформації про оренду клієнтам DHCP. Варто зазначити, що всі клієнтські операційні системи на базі Windows містять клієнтську частину DHCP як невід'ємний компонент стека протоколів TCP/IP, причому цей клієнт активовано в системі за замовчуванням.

Протокол динамічної конфігурації хоста (DHCP) – це протокол архітектури «клієнт-сервер», який автоматично забезпечує хост Інтернет-протоколу (IP) його IP-адресою та іншою супутньою конфігураційною інформацією, такою як маска підмережі та шлюз за замовчуванням. У регламентуючих документах RFC 2131 та 2132 протокол DHCP визначено як стандарт IETF (Internet Engineering Task Force), що базується на протоколі початкового завантаження (BOOTP) – протоколі, з яким DHCP має значну кількість спільних деталей реалізації, що дозволяє хостам отримувати необхідну конфігурацію TCP/IP від сервера DHCP [31].

Для забезпечення доступу до мережі та її ресурсів кожен пристрій у мережі на базі TCP/IP повинен володіти унікальною одноадресною (unicast) IP-адресою. За відсутності служби DHCP налаштування IP-адрес для нових комп'ютерів або комп'ютерів, що переміщуються з однієї підмережі в іншу, повинно виконуватися вручну. Аналогічним чином вручну повинно здійснюватися вивільнення IP-адрес для комп'ютерів, які вилучаються з мережі. Запровадження DHCP дозволяє повністю автоматизувати цей процес та здійснювати його централізовано. Сервер DHCP підтримує пул IP-адрес і надає адресу в оренду будь-якому клієнту з підтримкою DHCP під час його ініціалізації в мережі. Оскільки IP-адреси є динамічними (надаються в оренду), а не статичними (призначаються на постійній основі), адреси, що більше не використовуються, автоматично повертаються до пулу для подальшого перерозподілу [31].

Адміністратором мережі встановлюються сервери DHCP, які підтримують інформацію про конфігурацію TCP/IP і надають налаштування адреси клієнтам із підтримкою DHCP у формі пропозиції оренди. Сервер DHCP зберігає інформацію про конфігурацію в базі даних, яка включає дійсні параметри конфігурації TCP/IP для всіх клієнтів у мережі, а також дійсні IP-адреси, що утримуються в пулі для призначення клієнтам, та адреси, що були виключені. Окрім того, база даних містить зарезервовані IP-адреси, асоційовані з конкретними клієнтами DHCP, що дозволяє забезпечити послідовне призначення однієї IP-адреси одному клієнту. Також визначається тривалість оренди, тобто проміжок часу, протягом якого IP-адреса може використовуватися до моменту, коли вимагатиметься подовження оренди. Клієнт із підтримкою DHCP після прийняття пропозиції оренди отримує дійсну IP-адресу для підмережі, до якої він підключається, та запитані параметри DHCP. Останні являють собою додаткові параметри, які налаштовані на сервері DHCP для призначення клієнтам, наприклад, адреса маршрутизатора (шлюз за замовчуванням), DNS-сервери та доменне ім'я DNS [13].

Використання сервера DHCP забезпечує низку переваг, зокрема надійну конфігурацію IP-адрес. DHCP мінімізує помилки конфігурації, викликані ручним налаштуванням IP-адрес, такі як друкарські помилки або конфлікти адрес, спричинені призначенням однієї IP-адреси більш ніж одному комп'ютеру одночасно. Також досягається зменшення навантаження на адміністрування мережі завдяки таким функціям, як централізована та автоматизована конфігурація TCP/IP, можливість визначати конфігурації TCP/IP з центрального розташування та можливість призначати повний спектр додаткових значень конфігурації TCP/IP за допомогою параметрів DHCP [31].

Додатково забезпечується ефективна обробка змін IP-адрес для клієнтів, які потребують частого оновлення (наприклад, портативних пристроїв у бездротовій мережі), та пересилання початкових повідомлень DHCP за допомогою агента ретрансляції DHCP, що усуває потребу в розгортанні сервера DHCP у кожній підмережі.

Сервер DHCP у середовищі Windows Server включає розширені функціональні можливості, серед яких політики DHCP, що дозволяють створювати правила застосування параметрів на основі характеристик клієнта (наприклад, MAC-адреси або класу виробника), та журналювання аудиту DHCP для відстеження активності сервера, включаючи призначення та подовження оренди [31].

Керування серверами DHCP здійснюється за допомогою Windows PowerShell, консолі DHCP або Windows Admin Center. Важливими функціями є авторизація сервера DHCP в Active Directory для запобігання наданню адрес неавторизованими серверами, а також інтеграція з DNS, де динамічний DNS автоматично оновлює записи при зміні оренди. Передбачена інтеграція з протоколами IPv4 та IPv6 для підтримки обох стандартів адресації,

функція відмовостійкості, що дозволяє двом серверам спільно використовувати одну область для надлишковості та балансування навантаження, а також інтеграція з IPAM (IP Address Management) для централізованого керування призначеннями IP-адрес та орендою.

Типи повідомлень протоколу

Функціонування служби DHCP у середовищі Windows Server 2025 базується на суворо регламентованому обміні службовими повідомленнями, які інкапсулюються в дейтаграми транспортного протоколу UDP.

Архітектура взаємодії клієнта та сервера реалізується через кінцевий автомат станів, переходи між якими ініціюються отриманням або відправленням специфічних пакетів. Ключовим елементом ідентифікації типу повідомлення є поле опції 53 (DHCP Message Type), яке міститься в заголовку кожного пакета. Хоча базовий стандарт протоколу залишається незмінним (згідно з RFC 2131), реалізація стека TCP/IP у Windows Server 2025 забезпечує оптимізовану обробку цих повідомлень, підтримку розширених політик безпеки та інтеграцію з протоколом IPv6. Розрізняють вісім основних типів повідомлень для IPv4 та аналогічний, але термінологічно відмінний набір для IPv6, кожен з яких виконує критичну функцію в життєвому циклі оренди адреси [32].

Процес отримання мережевих налаштувань ініціюється повідомленням DHCPDISCOVER. Це широкомовний запит, що генерується клієнтом для локалізації доступних DHCP-серверів у межах фізичного сегмента мережі або за його межами через агенти ретрансляції. У Windows Server 2025 обробка цього повідомлення включає перевірку на відповідність політикам фільтрації MAC-адрес та наявність дозволів у списку дозволених/заборонених клієнтів (Allow/Deny) ще до моменту формування відповіді. Пакет надсилається на широкомовну адресу 255.255.255.255 із використанням порту джерела 68 та порту призначення 67. У структуру повідомлення часто включається опція 61 (Client Identifier – Ідентифікатор Клієнта) та запит на певні параметри (Опція 55), що дозволяє серверу заздалегідь визначити необхідну конфігурацію для конкретного пристрою [32].

У відповідь на валідний запит DHCPDISCOVER сервером генерується повідомлення DHCPOFFER. Цей пакет містить попередню пропозицію оренди, яка включає вільну IP-адресу з відповідної області, маску підмережі, тривалість оренди та ідентифікатор сервера. Варто зазначити, що у Windows Server 2025 механізм пропозиції адреси тісно інтегрований із службою захисту доступу до мережі, що дозволяє динамічно змінювати пропоновані параметри залежно від стану «здоров'я» клієнта. На цьому етапі запропонована IP-адреса тимчасово резервується сервером, щоб уникнути її видачі іншому клієнту до завершення транзакції. Повідомлення може надсилатися як unicast, так і broadcast, залежно від прапорця Broadcast у заголовку початкового запиту клієнта.

Важливим етапом узгодження параметрів є надсилання клієнтом повідомлення DHCPREQUEST. Цей тип повідомлення використовується в трьох різних сценаріях: для вибору конкретного сервера після отримання пропозиції DHCPOFFER (при цьому пропозиції інших серверів імпліцитно відхиляються), для підтвердження раніше отриманої адреси після перезавантаження системи, а також для періодичного подовження терміну дії оренди (Renewal). У середовищі Windows Server 2025, при налаштованій відмовостійкості, вміст цього повідомлення синхронізується між партнерами по відмовостійкості, що забезпечує безперервність обслуговування навіть у разі виходу з ладу основного вузла. При початковому виборі сервера повідомлення надсилається широкомовно, щоб повідомити всі інші сервери про те, що їхні пропозиції не були прийняті, що дозволяє їм повернути зарезервовані адреси до пулу доступних.

Успішне завершення процесу конфігурації фіксується відправленням повідомлення DHCPACK (від англ. Acknowledgement). Цей пакет містить фінальне підтвердження надання IP-адреси та повний набір опцій DHCP, таких як адреси DNS-серверів, шлюзу за замовчуванням, доменне ім'я та інші специфічні. Отримання цього повідомлення переводить клієнта у стан «Bound». У випадку неможливості задоволення запиту клієнта (наприклад, якщо запитувана адреса вже зайнята або клієнт перемістився в іншу логічну підмережу), сервер Windows Server 2025 генерує повідомлення DHCPNAK (Negative Acknowledgement). Це змушує клієнта негайно припинити використання адреси та ініціювати процес отримання

налаштувань заново (повернутися до стадії Discover) [32].

Окрім основного циклу DORA (DISCOVER – OFFER – REQUEST – ACK), протокол передбачає механізми діагностики та коректного завершення роботи. Повідомлення DHCPDECLINE надсилається клієнтом, якщо він виявляє конфлікт IP-адрес (наприклад, за допомогою ARP-запитів) перед використанням запропонованої адреси. Windows Server 2025 позначає таку адресу як «BAD_ADDRESS» і тимчасово вилучає її з обігу до втручання адміністратора або автоматичного очищення [13].

Для звільнення адреси використовується повідомлення DHCPRELEASE, яке дозволяє серверу негайно повернути адресу в пул. Також підтримується повідомлення DHCPINFORM, що застосовується клієнтами зі статичними адресами для отримання додаткових опцій без виділення IP-адреси.

Окремо слід виділити повідомлення протоколу DHCPv6, підтримка якого є важливою для сучасних мереж на базі Windows Server 2025. Це такі повідомлення, як Solicit (аналог Discover), Advertise (аналог Offer), Request (аналог Request) та Reply (аналог Ack), які забезпечують аналогічний функціонал у мережах з використанням IPv6.

Основні поняття DHCP: DHCP-сервер, DHCP-клієнт

Служба DHCP-сервера являє собою останню реалізацію сучасної системи автоматизованої мережевої адресації. Функціонал даної служби включає виконання всіх тих же операцій, що й служба BOOTP, проте додатково забезпечується можливість надання розширеної інформації клієнтам, які ініціюють запит на отримання IP-адреси.

Сервером DHCP видача IP-адреси клієнту реалізується за допомогою процедури, що складається з трьох етапів. На першому етапі клієнт DHCP завантажується та здійснює розсилку DHCP-запиту на отримання IP-адреси всім вузлам у локальній мережі. На другому етапі DHCP-сервер, що знаходиться в локальній мережі, отримує цей запит і виконує підготовку до відправлення IP-адреси даному клієнту у формі DHCP-оренди. На завершальному етапі, після визначення DHCP-сервером необхідної інформації із запиту клієнта, здійснюється видача клієнту DHCP-оренди IP-адреси, яка включає також додаткові параметри оренди, такі як маска підмережі, шлюз за замовчуванням та, найімовірніше, IP-адреса самого сервера [15].

Клієнтська служба DHCP – це служба на стороні клієнта, що виконує запит IP-адреси з мережі. Залежно від конфігурації мережевого адаптера системи, клієнтська служба DHCP може перебувати в активному стані або бути відключеною. У випадках, коли клієнтом використовується мережеве завантаження, служба може набувати вигляду клієнта BOOTP або PXE, керованого системою [15].

У середовищі Windows керування клієнтською службою DHCP здійснюється на основі конфігурації, що зберігається в операційній системі Microsoft, а також безпосередньо на кожному адаптері. У разі виявлення адаптером підключення до мережі та за умови налаштування IP-конфігурації на автоматичну адресацію, клієнтською службою ініціюється розсилка запиту IP-адреси. Коли з сервера буде отримано відповідь, інформація про оренду застосовується до відповідного адаптера, після чого стає можливим повноцінний мережевий обмін даними.

Одночасно з DHCP-орендою IP-адреси клієнтом отримується важлива додаткова інформація – термін дії оренди. Цей параметр визначає часовий проміжок, протягом якого клієнт має право використовувати видану IP-адресу до моменту необхідності повторного звернення до DHCP-сервера з метою подовження існуючої або отримання нової оренди.

Клієнтом DHCP здійснюється кешування цієї інформації. У ситуаціях, коли термін оренди наближається до завершення, або під час перезапуску системи чи нової ініціалізації мережі, клієнт DHCP встановлює зв’язок із DHCP-сервером для перевірки дійсності оренди або необхідності її подовження чи заміни.

Окрім зазначеного, у системах Microsoft на клієнтську службу DHCP покладається функція керування реєстрацією клієнта в динамічній DNS за умови доступності відповідного сервера динамічної DNS. Однак цей алгоритм не застосовується у випадку, коли служба DHCP-сервера передає реєстрацію DHCP-оренди в динамічній DNS безпосередньо самому серверу.

Тема 15
Основи Linux-адміністрування. Мережеві служби Linux. Веб-сервісна інфраструктура у Linux

Структура ОС Linux

Структура операційної системи Linux являє собою складну, багаторівневу ієрархічну структуру, яка спроектована для забезпечення ефективного керування апаратними ресурсами обчислювальної машини та надання абстрагованого інтерфейсу для прикладного програмного забезпечення. Фундаментальною основою побудови системи є чітке розмежування простору виконання на привілейований режим, відомий як простір ядра та режим користувача. Така сегрегація реалізується на апаратному рівні через механізм кілець захисту процесора, де ядро функціонує у «кільці 0», маючи повний доступ до пам'яті та периферії, тоді як прикладні програми та системні служби працюють у «кільці 3» з обмеженими правами. Взаємодія між цими двома просторами здійснюється виключно через суворо регламентований інтерфейс системних викликів, що гарантує стабільність роботи сервера. Збій у прикладному додатку не призводить до критичної зупинки всієї операційної системи.

Центральним компонентом системи виступає ядро Linux, яке класифікується як монолітне з можливістю динамічного завантаження модулів. Монолітність архітектури передбачає, що всі критично важливі підсистеми – планувальник процесів, менеджер пам'яті, драйвери пристроїв, мережевий стек та віртуальна файлова система – виконуються в єдиному адресному просторі, що забезпечує високу продуктивність завдяки мінімізації накладних витрат на перемикання контексту. Водночас модульний принцип дозволяє розширювати функціональність ядра без необхідності перезавантаження сервера, шляхом підключення зовнішніх об'єктних файлів. Ця особливість є критично важливою для серверних інфраструктур, де вимагається забезпечення безперервної роботи при зміні апаратної конфігурації або оновленні драйверів [36].

Важливою концептуальною особливістю Linux, успадкованою від систем UNIX, є парадигма «все є файл». Цей принцип уніфікації означає, що доступ до більшості системних ресурсів, включаючи дискові накопичувачі, мережеві сокети, канали міжпроцесної взаємодії та навіть інформацію про стан оперативної пам'яті, здійснюється через стандартні операції читання та запису файлових дескрипторів. Для реалізації цього підходу в ядрі спроектовано прошарок віртуальної файлової системи (VFS), який надає єдиний універсальний інтерфейс для роботи з різнорідними фізичними файловими системами (наприклад, ext4, XFS, Btrfs) та мережевими протоколами. Організація файлового простору підпорядковується стандарту ієрархії файлової системи (Filesystem Hierarchy Standard – FHS), що визначає призначення стандартних каталогів та забезпечує сумісність між різними дистрибутивами та програмним забезпеченням [36].

Управління життєвим циклом системи покладається на підсистему ініціалізації, яка запускається безпосередньо після завантаження ядра і отримує ідентифікатор процесу PID 1. У сучасних дистрибутивах Linux, в тому числі і серверних версіях ОС, цю роль виконує системний менеджер systemd, який замінив застарілу модель SysVinit. Systemd спроектовано для забезпечення паралельного запуску служб, відстеження залежностей між компонентами, керування логуванням через journald та автоматичного перезапуску критичних демонів у разі їх аварійної зупинки. Розуміння механізмів роботи процесу ініціалізації є необхідним для адміністратора, оскільки саме на цьому рівні конфігурується автозавантаження веб-серверів, баз даних та інших мережевих служб, що формують серверну інфраструктуру.

Інтерфейс взаємодії адміністратора з системою реалізується, в основному, через командну оболонку, яка виступає інтерпретатором команд та середовищем для виконання скриптів автоматизації. Оболонка функціонує у просторі користувача і не є частиною ядра, що дозволяє використовувати різні варіанти інтерпретаторів (Bash, Zsh, Sh) залежно від потреб. У серверному середовищі Linux графічний інтерфейс зазвичай відсутній з метою економії ресурсів та підвищення безпеки, тому основна робота з налаштування мережевих служб та моніторингу продуктивності виконується через термінальний доступ.

Багатокористувацька природа ОС Linux забезпечується системою прав доступу на основі ідентифікаторів користувачів (UID) та груп (GID), а також атрибутів файлів, що дозволяє чітко розмежувати права на читання, запис та виконання для різних суб'єктів системи, забезпечуючи конфіденційність та цілісність даних у багатокористувацькому середовищі.

Специфіка серверної структури Linux полягає в орієнтації на забезпечення максимальної продуктивності та надійності при обробці конкурентних мережевих запитів, що досягається шляхом використання спеціалізованих профілів планувальника завдань та відмовою від графічної підсистеми хоча є можливість і встановлення графічного компонента для зручності адміністрування.

Робота сервера в режимі «headless» (без монітора та периферії введення) дозволяє вивільнити критичні системні ресурси – процесорний час та оперативну пам'ять – для обслуговування веб-серверів, систем керування базами даних або балансувальників навантаження, замість відмальовування інтерфейсу користувача. Налаштування параметрів ядра через механізм sysctl у серверних дистрибутивах зазвичай виконується з пріоритетом на пропускну здатність мережевого стека та ефективність дискового введення-виводу, на відміну від десктопних систем, де пріоритетом є мінімізація затримок для комфорту користувача. Управління такою інфраструктурою здійснюється дистанційно через захищені криптографічні протоколи, зокрема SSH, що вимагає впровадження суворих політик безпеки та використання засобів автоматизації конфігурування, оскільки фізичний доступ до обладнання в дата-центрах часто є неможливим [36].

Принципи роботи з командним рядком

Взаємодія адміністратора з серверною операційною системою Ubuntu Server здійснюється переважно через командний рядок (Command Line Interface – CLI), який виступає основним інструментом керування системними ресурсами, конфігурацією служб та діагностикою мережевої інфраструктури. На відміну від графічних інтерфейсів, які приховують складність внутрішніх процесів за візуальними абстракціями, командний рядок забезпечує безпосередній доступ до системних викликів та утиліт, дозволяючи виконувати операції з максимальною точністю та ефективністю.

У середовищі Ubuntu Server за замовчуванням використовується командна оболонка Bash (Bourne Again Shell), яка функціонує як інтерпретатор командної мови. Вона зчитує текстові команди, введені користувачем, здійснює їх синтаксичний аналіз та ініціює виконання відповідних програм або системних процедур. Розуміння принципів роботи оболонки є критичним для автоматизації адміністративних завдань, оскільки Bash підтримує змінні, умовні оператори, цикли та функції, перетворюючи командний рядок на повноцінне середовище програмування сценаріїв.

Синтаксична структура команд у середовищі Linux підпорядковується суворому формату, що зазвичай складається з назви утиліти (команди), опцій (ключів або прапорців), які модифікують поведінку програми, та аргументів, що визначають об'єкт дії (файл, каталог, процес або мережевий хост). Процес виконання команди розпочинається з пошуку відповідного виконуваного файлу в файловій системі. Оболонка використовує змінну оточення PATH, яка містить впорядкований перелік директорій, де здійснюється пошук бінарних файлів. Якщо команда не є вбудованою функцією оболонки (наприклад, cd або echo) і не знайдена в шляхах, визначених у змінній PATH, система повертає повідомлення про помилку. Важливою особливістю роботи з аргументами є чутливість файлової системи Linux до регістру символів, що вимагає точності при введенні назв файлів та каталогів, а також використання механізму автодоповнення (tab completion) для підвищення продуктивності та уникнення синтаксичних помилок [36].

Фундаментальною концепцією роботи в командному рядку Unix-подібних систем є модель стандартних потоків введення-виведення, яка забезпечує універсальний механізм обміну даними між процесами. Кожна запущена програма автоматично асоціюється з трьома дескрипторами потоків: стандартним введенням (stdin, дескриптор 0), стандартним виведенням (stdout, дескриптор 1) та стандартним потоком помилок (stderr, дескриптор 2).

За замовчуванням введення здійснюється з клавіатури, а виведення спрямовується на термінал користувача. Проте, архітектура оболонки дозволяє змінювати напрямок цих

потоків за допомогою операторів перенаправлення. Це дозволяє зберігати результати роботи програм у файли, ігнорувати повідомлення про помилки або зчитувати вхідні дані з попередньо підготовлених джерел, що є необхідним для ведення логування та автоматичного виконання завдань без участі оператора.

Особливу потужність командному рядку надає механізм конвеєризації, який реалізується за допомогою символу вертикальної риски (|). Конвеєр дозволяє об'єднувати прості, спеціалізовані утиліти в складні ланцюжки обробки даних, передаючи стандартне виведення однієї команди безпосередньо на стандартне введення іншої. Цей принцип, відомий як філософія Unix («роби одну річ, але роби її добре»), дозволяє адміністратору виконувати складні маніпуляції з текстовими даними, фільтрацію логів, сортування процесів та обробку мережевого трафіку в реальному часі без необхідності написання спеціалізованого програмного забезпечення. Наприклад, комбінація утиліт для перегляду вмісту файлів, пошуку за шаблоном (grep), сортування (sort) та підрахунку унікальних входжень (uniq) є стандартним патерном для аналізу журналів доступу веб-сервера.

Адміністрування Ubuntu Server вимагає чіткого розуміння моделі привілеїв та керування сеансами користувачів. Оскільки пряме використання облікового запису суперкористувача (root) вважається небезпечною практикою, що підвищує ризик випадкового пошкодження системи, в Ubuntu застосовується механізм sudo (SuperUser DO). Цей інструмент дозволяє делегувати адміністративні повноваження звичайним користувачам, вимагаючи підтвердження особистості через введення пароля та логуючи кожну виконану команду з підвищеними правами. Робота в командному рядку також передбачає керування фоновими та активними процесами. Адміністратор має можливість переводити тривалі операції у фоновий режим, призупиняти або завершувати їх за допомогою сигналів, що надсилаються процесам через утиліти керування завданнями або команду kill, забезпечуючи таким чином гнучкий контроль за навантаженням на сервер.

Оскільки доступ до сервера найчастіше здійснюється віддалено, принципи роботи з командним рядком нерозривно пов'язані з використанням протоколу SSH. Цей протокол забезпечує захищений, шифрований канал зв'язку для передачі команд та отримання результатів їх виконання через незахищені мережі. При роботі через SSH командна оболонка функціонує на віддаленому сервері, але відображення результатів відбувається на локальному терміналі клієнта. Це накладає певні особливості на роботу з інтерактивними програмами та вимагає використання термінальних мультиплексорів (наприклад, tmux або screen), які дозволяють зберігати сеанс роботи активним навіть у разі розриву мережевого з'єднання, що є критично важливим при виконанні тривалих оновлень системи або компіляції програмного забезпечення на віддаленому сервері.

Налаштування мережевих параметрів

Конфігурування мережевих інтерфейсів є одним з основних етапів розгортання серверної інфраструктури, оскільки коректна адресація забезпечує доступність сервісів для клієнтів та взаємодію з іншими вузлами мережі. В операційній системі Linux, зокрема в дистрибутиві Ubuntu Server, керування мережею базується на взаємодії ядра з простором користувача через спеціалізовані демони, такі як NetworkManager або systemd-networkd. Для серверних систем критично важливим є використання статичної IP-адресації замість динамічної (DHCP), оскільки це гарантує постійність точки входу для веб-серверів, баз даних та служб віддаленого доступу. Процес налаштування вимагає розуміння параметрів протоколу IPv4, зокрема IP-адреси хоста, маски підмережі, яка визначає межі широкомовної ділянки мережі, та шлюзу за замовчуванням, що забезпечує маршрутизацію трафіку до зовнішніх мереж.

Алгоритм процесу налаштування статичної адресації можна формалізувати як послідовність логічно пов'язаних етапів. На початковому кроці здійснюється ідентифікація активного мережевого інтерфейсу та отримання прав доступу до зміни його конфігурації. Другий етап передбачає переведення методу отримання адреси з автоматичного режиму в ручний, що виключає вплив DHCP-сервера на параметри хоста. Третій етап полягає у безпосередньому введенні мережевих реквізитів (адреса, маска, шлюз, DNS-сервери) у відповідні конфігураційні файли або графічні форми. Четвертий етап вимагає перезапуску

мережевої служби або інтерфейсу для ініціалізації нових параметрів ядром системи. Завершальним етапом є верифікація з'єднання шляхом надсилання ехо-запитів до шлюзу та зовнішніх вузлів [36].

При використанні серверних дистрибутивів ОС із встановленим графічним оточенням, налаштування часто виконується через графічний інтерфейс користувача, що дозволяє візуалізувати параметри NetworkManager. Процедура розпочинається із запуску екземпляра Linux Server, розгорнутого в результаті виконання попередніх етапів проектування інфраструктури. Після завантаження операційної системи здійснюється автентифікація в системі під обліковим записом користувача, що має відповідні права. Налаштування ініціюється відкриттям утиліти системних параметрів («Settings»), де необхідно обрати розділ керування мережею («Network»). У переліку доступних апаратних інтерфейсів обирається провідне з'єднання («Wired»), після чого здійснюється перехід до режиму редагування параметрів шляхом натискання на піктограму конфігурації (шестерні), як це продемонстровано на рисунку 15.1.

Внаслідок виконання вищезазначених дій ініціюється вікно налаштувань конкретного підключення, що містить деталізовані параметри мережевого адаптера. Для зміни схеми адресації необхідно здійснити перехід у вкладку «IPv4» у вікні, що відкрилося, яка відповідає за конфігурацію протоколу Інтернет четвертої версії (рис. 15.2).

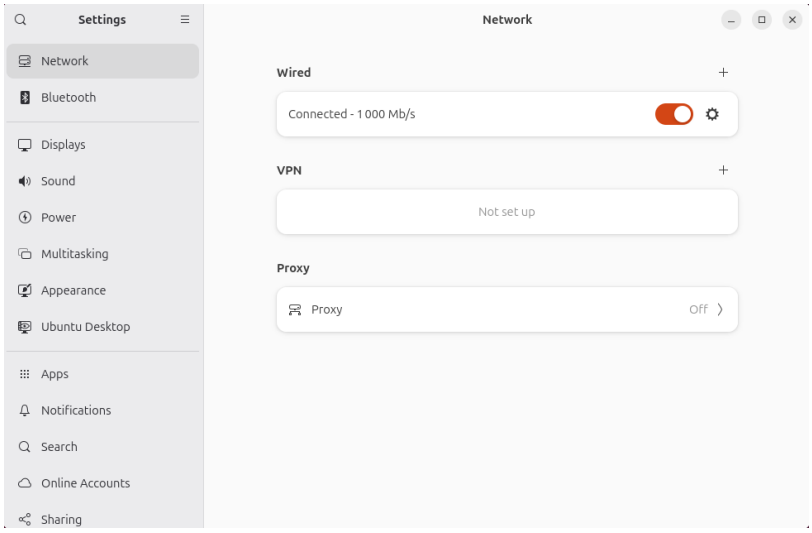

Рисунок 15.1 – Налаштування мережі [36]

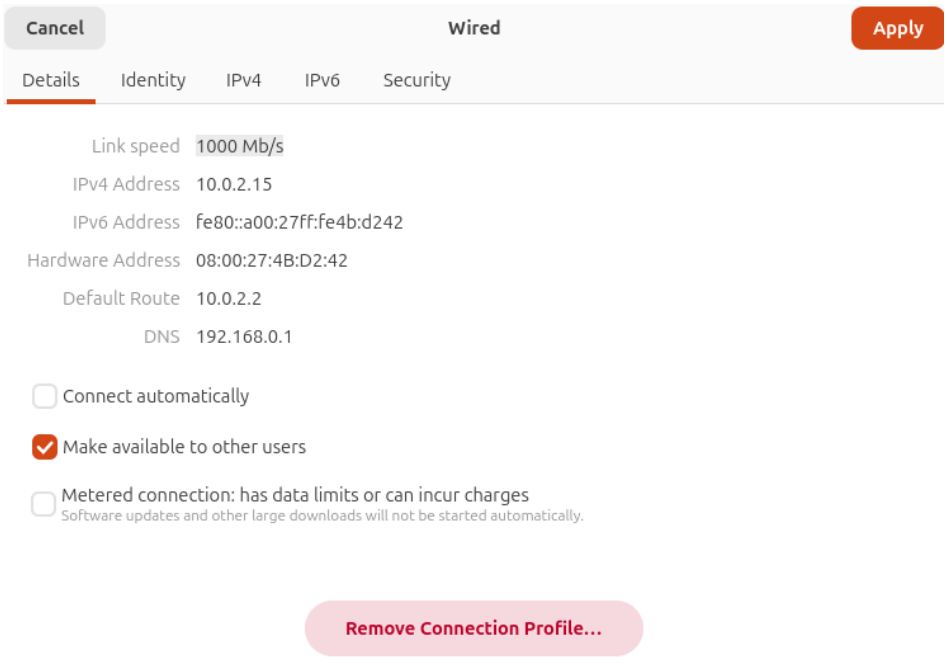

Рисунок 15.2 – Налаштування провідного з'єднання мережі [36]

У межах вкладки «IPv4» здійснюється ключова зміна логіки роботи інтерфейсу: у полі вибору методу присвоєння адреси («Method») встановлюється значення «Manual» (Ручний), що деактивує клієнт DHCP. Далі у секції «Addresses» виконується введення детермінованих мережевих параметрів, визначених топологією мережі. Зокрема, у відповідні поля вносяться наступні значення: IP-адреса хоста – наприклад, 192.168.56.10; маска підмережі – наприклад, 255.255.255.0, що відповідає префіксу /24; та шлюз за замовчуванням – наприклад, 192.168.56.1. Фіксація внесених змін здійснюється натисканням кнопки «Apply» (Застосувати). Для того щоб нові параметри набули чинності на рівні ядра, необхідно виконати реініціалізацію інтерфейсу шляхом його вимкнення та повторного ввімкнення засобами графічного інтерфейсу (рис. 15.3)

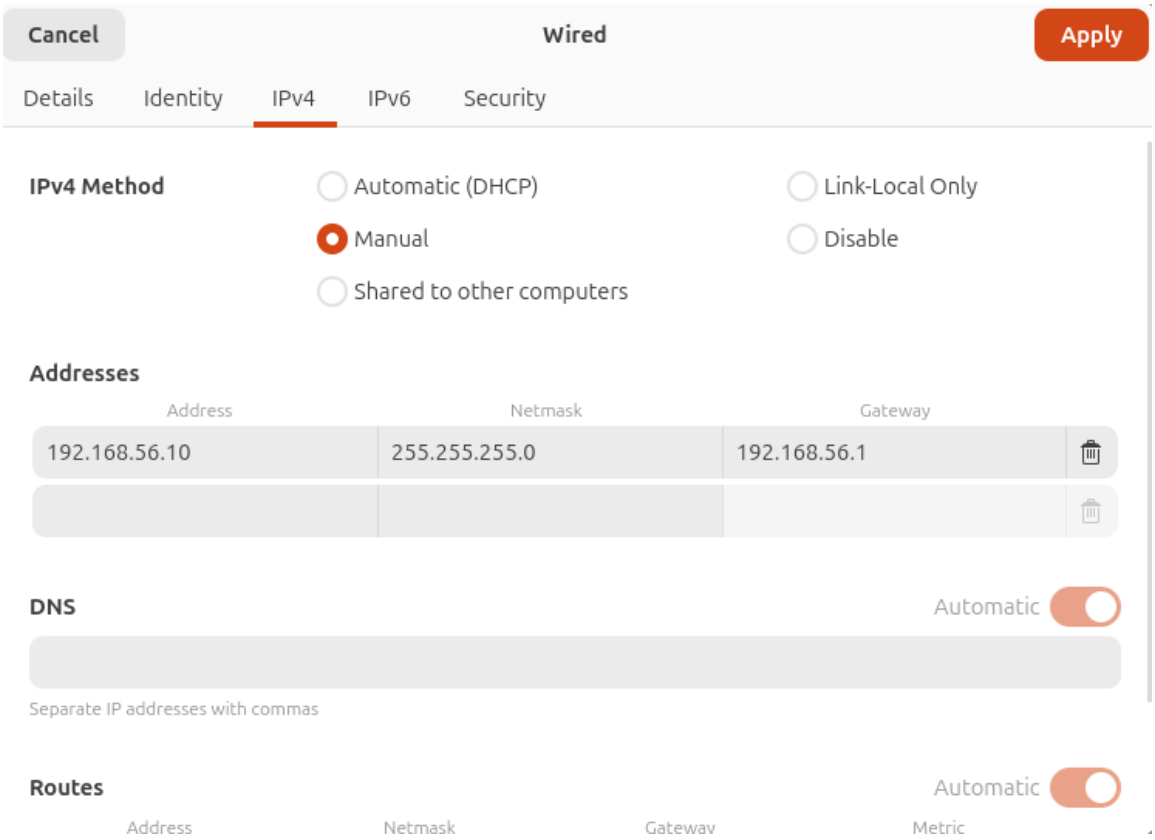

Рисунок 15.3 – Налаштування статичної IP-адресації для сервера [36]

Завершальним етапом конфігурування є верифікація застосованих параметрів та перевірка доступності вузлів мережі. Для діагностики використовується емулятор термінала, в якому виконуються команди перевірки мережевого стека. Команда ip addr show дозволяє пересвідчитися, що інтерфейсу дійсно присвоєно статичну адресу 192.168.56.10.

Наступним кроком виконується перевірка зв'язності з локальним шлюзом за допомогою команди ping -c 4 192.168.56.1, яка надсилає чотири ICMP-пакети. Аналогічна перевірка здійснюється для зовнішнього вузла (наприклад, DNS-сервера Google) командою ping -c 4 8.8.8.8. Успішне отримання відповідей на ехо-запити свідчить про коректність налаштування статичної IP-адресації та працездатність маршрутизації.

Управління користувачами

Основою безпеки та організації роботи в операційній системі Linux, і зокрема в дистрибутиві Ubuntu Server, є концепція багатокористувацького середовища. Архітектура системи спроектована таким чином, що кожен процес, який виконується в просторі користувача, повинен бути асоційований з певним суб'єктом – обліковим записом користувача. Цей підхід дозволяє ядру операційної системи реалізовувати механізми розмежування доступу, ізоляції процесів та аудиту дій.

З точки зору ядра, користувач ідентифікується не за текстовим іменем (логіном), а за унікальним числовим ідентифікатором – UID (User Identifier). Аналогічним чином, для групування користувачів та надання їм спільних прав доступу використовується

ідентифікатор групи – GID (Group Identifier). Процес трансляції зрозумілих людині текстових імен у числові ідентифікатори та навпаки здійснюється системними бібліотеками шляхом звернення до спеціалізованих баз даних облікових записів [36].

Центральним сховищем інформації про користувачів у системі є файл /etc/passwd. Цей файл являє собою текстову базу даних, де кожен рядок описує окремий обліковий запис і складається з семи полів, розділених двокрапкою. Перше поле містить логін користувача, який використовується при автентифікації. Друге поле історично призначалося для зберігання хешу пароля, проте в сучасних системах, з міркувань безпеки, там розміщується символ «x», що вказує на перенесення криптографічних даних у захищений файл /etc/shadow. Третє та четверте поля містять, відповідно, UID користувача та GID його основної групи. П'яте поле, відоме як GECOS, використовується для зберігання додаткової інформації, такої як повне ім'я користувача, номер кабінету чи телефон. Шосте поле визначає абсолютний шлях до домашнього каталогу користувача, де зберігаються його особисті файли та налаштування. Сьоме поле вказує на командну оболонку, яка буде запущена автоматично після успішного входу в систему. Якщо в цьому полі вказати /sbin/nologin або /bin/false, вхід для даного користувача буде заблоковано, що часто використовується для сервісних облікових записів.

Важливим аспектом адміністрування є розуміння ієрархії користувачів. Користувач з UID 0 (root) має необмежені права доступу до всіх ресурсів системи, ігноруючи будь-які перевірки прав доступу. Діапазон UID від 1 до 999 (у більшості сучасних дистрибутивів, включаючи Ubuntu) зарезервовано для системних користувачів – спеціальних облікових записів, від імені яких запускаються системні служби (наприклад, www-data для веб-сервера, mysql для бази даних). Це забезпечує принцип найменших привілеїв: якщо зловмисник скомпрометує веб-сервер, він отримає права лише користувача www-data, а не адміністратора системи. Облікові записи реальних користувачів-людей зазвичай починаються з UID 1000. Коректне управління цими ідентифікаторами є необхідним при налаштуванні спільних файлових сховищ (NFS) або перенесенні архівів між різними серверами для уникнення колізій прав доступу [36].

Безпека автентифікації забезпечується механізмом тіньових паролів, реалізованим у файлі /etc/shadow. Цей файл доступний для читання виключно користувачу root, що унеможливлює проведення атак типу «brute-force» звичайними користувачами системи. У файлі зберігається не сам пароль, а його хеш-сума, отримана за допомогою криптостійких алгоритмів (в Ubuntu Server за замовчуванням використовується SHA-512, що позначається префіксом 6). Окрім хешу, файл містить параметри політики старіння паролів: дату останньої зміни, мінімальний та максимальний термін дії пароля, період попередження про необхідність зміни та час неактивності, після якого обліковий запис блокується. Адміністратор має можливість керувати цими параметрами за допомогою команди chage, примушуючи користувачів регулярно оновлювати паролі для підвищення загального рівня безпеки інфраструктури.

Процес створення нового користувача в Ubuntu Server може виконуватися за допомогою низькорівневої утиліти useradd або високорівневого скрипта adduser. Використання useradd надає адміністратору повний контроль над процесом, дозволяючи вручну вказати UID, GID, домашній каталог та інші параметри, але вимагає чіткого розуміння ключів запуску.

Натомість adduser, який є інтерактивною надбудовою над useradd у системах Debian/Ubuntu, автоматизує більшість рутинних операцій: він створює домашній каталог, копіює в нього конфігураційні файли за замовчуванням, пропонує ввести пароль та інформацію GECOS (рис. 15.4). При створенні домашнього каталогу система використовує шаблон, що знаходиться в директорії /etc/skel. Усі файли (зазвичай це приховані налаштування оболонки, такі як .bashrc та .profile), які розміщені в /etc/skel, автоматично копіюються до домашньої папки нового користувача, забезпечуючи стандартизоване початкове оточення.

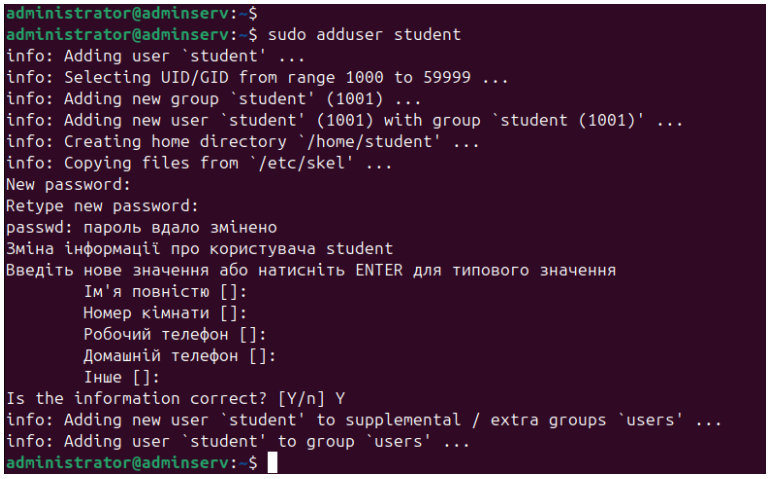

Рисунок 15.4 – Створення нового користувача з використанням adduser [36]

Керування групами є невід'ємною складовою моделі доступу. Кожен користувач повинен мати одну основну групу (primary group), GID якої записується в /etc/passwd. При створенні файлів новим користувачем, ці файли автоматично отримують групу-власника, що відповідає основній групі користувача. Окрім основної, користувач може бути членом довільної кількості додаткових груп (supplementary groups), інформація про які зберігається у файлі /etc/group. Це дозволяє гнучко налаштовувати доступ до проектних директорій або системних функцій. Наприклад, в Ubuntu членство в групі sudo надає адміністративні права, а група docker дозволяє керувати контейнерами без використання sudo. Модифікація членства в групах здійснюється командою usermod з ключами -aG (append group), де критично важливо використовувати ключ додавання -a, щоб не перезаписати існуючий список груп користувача (рис. 15.5).

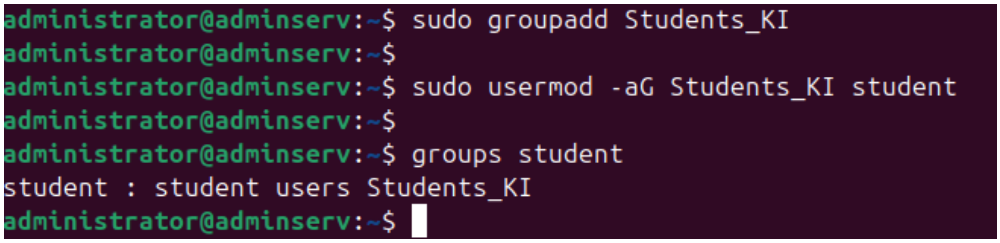

Рисунок 15.5 – Робота з групами в Ubuntu Server [36]

В Ubuntu Server за замовчуванням обліковий запис root заблокований (не має встановленого пароля), що унеможливлює прямий вхід в систему під цим ім'ям. Замість цього використовується механізм делегування повноважень sudo (SuperUser DO). Цей підхід значно підвищує безпеку, оскільки дозволяє адміністраторам виконувати привілейовані команди, використовуючи власний пароль, що забезпечує персоніфікований аудит дій у системних журналах. Конфігурація прав доступу sudo визначається у файлі /etc/sudoers. Редагування цього файлу повинно виконуватися виключно за допомогою спеціалізованої команди visudo, яка здійснює перевірку синтаксису перед збереженням.

Помилка в синтаксисі /etc/sudoers, допущена при прямому редагуванні текстовим редактором, може призвести до повної втрати можливості адміністрування системи. Синтаксис файлу дозволяє налаштовувати права з високою точністю, оскільки можна дозволити конкретному користувачу виконувати лише певний набір команд від імені root або іншого користувача, і навіть без запиту пароля.

Основою системи захисту файлової системи Linux є модель дискреційного розмежування доступу (Discretionary Access Control – DAC). Для кожного файлу або каталогу в файловій системі зберігаються атрибути, що визначають права доступу для трьох категорій суб'єктів: власника файлу (user), групи-власника (group) та всіх інших користувачів (others). Права визначаються трьома базовими дозволами: читання (r – read), запис (w – write) та виконання (x – execute). Інтерпретація цих прав відрізняється для файлів та каталогів. Для

файлу право r дозволяє зчитати його вміст, w – змінити вміст, а x – запустити файл як програму. Для каталогу право r дозволяє отримати список файлів у ньому, w – створювати або видаляти файли в каталозі, а x – входити в каталог (робити його поточним) та отримувати доступ до метаданих файлів у ньому [36].

Управління правами доступу здійснюється за допомогою команди chmod (change mode), яка підтримує два режими нотації: символьний та вісімковий (октальний) (рис. 15.6). У вісімковій системі правам присвоюються числові ваги: читання – 4, запис – 2, виконання – 1. Сума цих значень формує цифру прав для кожної категорії користувачів. Наприклад, комбінація 754 (rwxr-xr--) означає повні права для власника, права лише на читання і виконання для групи та виключно на читання для інших. Зміна власника файлу виконується командою chown (change owner), а групи – chgrp (change group) (рис. 15.7). Важливим поняттям є umask (user mask) – маска режиму створення файлів користувача, яка визначає права доступу за замовчуванням для новостворених об'єктів. Значення маски віднімається від базових повних прав, що дозволяє адміністратору глобально обмежувати права на нові файли для забезпечення конфіденційності.

```
administrator@adminserv:~$ touch test.txt
administrator@adminserv:~$
administrator@adminserv:~$ chmod 754 test.txt
administrator@adminserv:~$ chmod u=rwx,g=rwx,o=r test.txt
administrator@adminserv:~$
administrator@adminserv:~$ ls -l test.txt
-rwxrwxr-- 1 administrator administrator 0 сер 22 14:18 test.txt
administrator@adminserv:~$
```

Рисунок 15.6 – Управління правами доступу в Ubuntu Server двома режимами нотації [36]

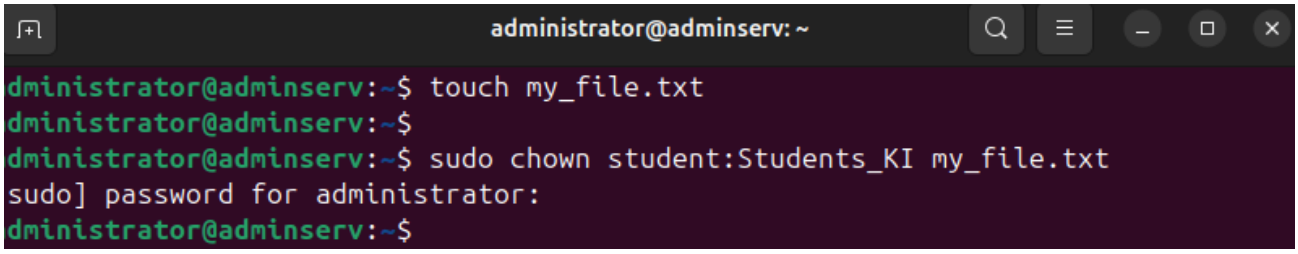

Рисунок 15.7 – Зміна власника файлу в Ubuntu Server [36]

Крім стандартних прав rwx, Linux підтримує спеціальні біти доступу: SUID (Set User ID), SGID (Set Group ID) та Sticky Bit. Встановлення біта SUID на виконуваний файл дозволяє звичайному користувачеві запускати програму з правами власника файлу (зазвичай root). Це механізм, який використовують такі утиліти, як passwd або ping, що потребують привілейованого доступу до системних ресурсів. Біт SGID на каталозі забезпечує спадкування групи-власника, тобто нові файли, створені в такому каталозі, отримують групу каталогу, а не основну групу користувача, що є ключовим для організації спільних робочих просторів. Sticky Bit встановлюється на каталоги загального доступу (наприклад, /tmp) і забороняє користувачам видаляти файли, власниками яких вони не є, навіть якщо вони мають право запису в каталог. У випадках, коли стандартної моделі прав недостатньо, застосовуються списки контролю доступу (Access Control Lists – ACL), що дозволяють надавати права на конкретний файл довільній кількості окремих користувачів або груп за допомогою команд setfacl та getfacl.

Принципи безпеки у Linux і відмінність безпеки між Linux і Windows

Забезпечення інформаційної безпеки в операційній системі Linux, зокрема в дистрибутиві Ubuntu Server, базується на комплексній, багаторівневій архітектурі захисту, яка охоплює рівень ядра, простору користувача, файлової системи та мережевих інтерфейсів.

Основним принципом побудови безпечної серверної інфраструктури є концепція

«глибокого захисту», яка передбачає створення декількох ешелонів захисту, де компрометація одного рівня не призводить до автоматичного отримання зловмисником повного контролю над системою. У середовищі Linux базовою аксіомою є принцип найменших привілеїв, згідно з яким будь-який процес або користувач повинен мати лише той мінімальний набір прав, який необхідний для виконання поставленого завдання. Цей підхід реалізується як через дискреційні механізми доступу (DAC), розглянуті в попередньому розділі цієї лекції, так і через мандатний контроль доступу (MAC).

Одним із ключових механізмів підвищення безпеки в сучасних дистрибутивах Linux, включаючи Ubuntu, є використання систем примусового контролю доступу (Mandatory Access Control – MAC), таких як AppArmor (Application Armor) або SELinux (Security-Enhanced Linux) [36].

На відміну від стандартної моделі DAC, де власник файлу самостійно визначає права доступу, модель MAC накладає централізовані політики безпеки, які не можуть бути змінені користувачами або навіть скомпрометованими програмами, що запущені з правами суперкористувача. В Ubuntu Server за замовчуванням використовується AppArmor, який працює на основі профілів безпеки, прив'язаних до конкретних виконуваних файлів. Ці профілі чітко регламентують, до яких файлів програма може звертатися, які мережеві порти відкривати та які системні виклики здійснювати. У випадку, якщо вразливість у веб-сервері (наприклад, Nginx) дозволить зловмиснику виконати довільний код, політика AppArmor заблокує спробу доступу до системних файлів, таких як /etc/shadow, навіть якщо процес веб-сервера теоретично мав би відповідні права власності, тим самим локалізуючи інцидент.

Мережева безпека серверного варіанту ОС Linux реалізується безпосередньо в ядрі операційної системи за допомогою підсистеми Netfilter, яка надає інфраструктуру для перехоплення та маніпуляції мережевими пакетами. Адміністрування правил фільтрації трафіку здійснюється через утиліти iptables або більш сучасний nftables.

Для спрощення конфігурації міжмережевого екрану в Ubuntu Server застосовується надбудова UFW (Uncomplicated Firewall), яка дозволяє адміністратору оперувати високорівневими поняттями, такими як дозвіл або заборона трафіку на конкретних портах чи для певних сервісів. Стратегія налаштування брандмауера повинна базуватися на принципі «заборонити все, що не дозволено явно» (default deny). Це означає, що початкова політика для вхідного трафіку встановлюється у значення DROP або REJECT, після чого додаються виключення лише для критично необхідних портів (наприклад, 22 для SSH, 80/443 для HTTP/HTTPS). Крім того, на рівні ядра налаштовуються параметри захисту від атак типу «відмова в обслуговуванні» (DoS), такі як SYN-cookies, відключення перенаправлення пакетів (IP forwarding) для серверів, що не є маршрутизаторами, та ігнорування ICMP-запитів.

Захист віддаленого доступу до сервера є теж важливим аспектом безпеки сервера, оскільки протокол SSH є основним вектором для спроб несанкціонованого проникнення. Стандартна конфігурація SSH вимагає значного посилення. Першочерговим заходом є повна заборона входу для користувача root через SSH, що змушує адміністратора входити під звичайним обліковим записом і підвищувати привілеї через sudo, залишаючи слід в аудиті. Далі рекомендується відмовитися від автентифікації за паролем на користь використання пар криптографічних ключів (RSA, ED25519), що унеможливлює успішне проведення атак методом підбору пароля (brute-force). Додатковим рівнем захисту є зміна стандартного порту прослуховування 22 на нестандартний для зменшення кількості автоматизованих сканувань, а також використання інструментів типу Fail2Ban, які аналізують логи (журнали) автентифікації в реальному часі та динамічно додають правила в брандмауер для блокування IP-адрес, з яких фіксуються багаторазові невдалі спроби входу [36].

У контексті безпеки веб-сервісної інфраструктури (Apache, Nginx) та файлових служб (Samba, NFS) застосовуються специфічні механізми ізоляції. Одним із найефективніших методів є запуск служб у середовищі chroot (change root), яке змінює видимий кореневий каталог для процесу, ізолюючи його від основної файлової системи. Це гарантує, що навіть у разі зламу веб-сервісу зловмисник не зможе вийти за межі визначеного дерева каталогів. Також важливою є регулярна перевірка конфігурації SSL/TLS – відключення застарілих

версій протоколів (SSLv3, TLS 1.0, 1.1) та слабких шифрів, налаштування HSTS (HTTP Strict Transport Security) для запобігання атакам зі зниженням рівня захисту.

Система аудиту та журналювання подій у Linux виконує роль «чорної скриньки», дозволяючи реконструювати послідовність подій у разі інциденту з безпекою. Центральним елементом журналювання є rsyslog або journald (у системах із systemd), які збирають повідомлення від ядра та служб. Для забезпечення цілісності журналів на серверах із високими вимогами до безпеки налаштовується відправка логів на віддалений виділений сервер у реальному часі. Це гарантує, що зловмисник, отримавши доступ до сервера, не зможе знищити сліди своєї діяльності, оскільки копії журналів вже будуть збережені на іншому вузлі. Додатково використовується підсистема Linux Audit Framework (auditd), яка дозволяє налаштувати детальне відстеження системних викликів, доступу до конкретних файлів або зміни атрибутів, надаючи значно детальнішу інформацію, ніж стандартні логи.

Проведення порівняльного аналізу безпеки серверних операційних систем Linux та Windows вимагає розгляду архітектурних відмінностей, моделей управління доступом та екосистем програмного забезпечення. Першою суттєвою відмінністю є модель розробки, адже Linux базується на відкритому вихідному коді, тоді як Windows є пропрієтарною системою із закритим кодом.

Архітектурна відмінність також полягає в способі зберігання конфігурації. У Linux усі налаштування зберігаються у текстових файлах, що спрощує їх перевірку, версіонування (через Git) та автоматизований аудит за допомогою скриптів. У Windows конфігурація зосереджена у Реєстрі – ієрархічній базі даних, яка є бінарною, менш прозорою та складнішою для аналізу змін без спеціалізованих інструментів. Крім того, Linux є модульною системою, де адміністратор може і повинен видалити всі непотрібні компоненти, зменшуючи поверхню атаки. Ядро Windows є більш монолітним у своїй комерційній поставці, і хоча існують варіанти Windows Server Core або Nano Server, стандартні інсталяції часто містять значну кількість успадкованих компонентів та графічний інтерфейс.

Суттєві відмінності спостерігаються в системах контролю доступу. Windows використовує складну систему списків контролю доступу (ACL) для NTFS та об’єктів Active Directory, яка забезпечує надзвичайно високу гранулярність прав (наприклад, окремі права на створення папок, запис атрибутів, зміну дозволів), але є складною в адмініструванні та діагностиці конфліктів. Linux використовує простішу модель rwx (читання/запис/виконання), доповнену списками ACL POSIX та атрибутами. Модель Windows є більш гнучкою та кращою в сфері систем контролюю доступу, проте модель Linux у поєднанні з SELinux/AppArmor надає дещо кращі можливості для ізоляції процесів. У Windows механізм UAC (User Account Control) виконує функцію, аналогічну sudo в Linux [13].

Окремо слід виділити питання шкідливого програмного забезпечення. Статистично, переважна більшість вірусів, троянів та програм-вимагачів створюється для середовища Windows. Це зумовлено як домінуванням Windows на десктопному ринку, так і можливістю виконання бінарних файлів (.exe), завантажених з довільних джерел в Інтернеті. У Linux інсталяція програмного забезпечення здійснюється переважно з довірених репозиторіїв пакетів (APT, YUM), які підписуються цифровими ключами розробників дистрибутиву. Це створює «ланцюг довіри», який значно ускладнює розповсюдження шкідливого ПЗ. Для запуску вірусу в Linux користувач часто повинен свідомо надати файлу права на виконання та ввести пароль адміністратора, що робить випадкове зараження малоймовірним. Проте, Linux-сервери є пріоритетною ціллю для складних таргетованих атак, спрямованих на веб-сервіси та бази даних, а не на саму ОС і в цьому поступаються Windows Server.

В аспекті оновлень та патч-менеджменту підхід Linux дозволяє оновлювати всі встановлені компоненти системи (ядро, бібліотеки, прикладне ПЗ) однією командою через пакетний менеджер. Windows Server має централізовану службу Windows Update для швидкого виконання оновлень.

Також Windows пропонує більш потужні інструменти централізованого управління політиками безпеки через об’єкти групових політик (GPO) в середовищі Active Directory, що є беззаперечним стандартом для корпоративних мереж, тоді як в Linux аналогічний рівень централізації досягається використанням інструментів конфігураційного управління (Ansible,

Puppet, Chef) або LDAP-рішень (FreeIPA), які вимагають вищої кваліфікації для впровадження, є більш складними у всіх аспектах та не перевершують можливості Windows Server [13].

Встановлення та налаштування DHCP і DNS у Linux

Протокол динамічної конфігурації хостів (DHCP) є надважливим елементом мережевої інфраструктури, що забезпечує автоматизований розподіл IP-адрес та супутніх параметрів (маски підмережі, шлюзу за замовчуванням, DNS-серверів) клієнтським пристроям. В ОС Linux найбільш поширеним та стабільним рішенням для реалізації DHCP-сервера є пакет ISC DHCP Server, розроблений Internet Systems Consortium. Його використання дозволяє мінімізувати ймовірність виникнення конфліктів IP-адрес та значно спрощує адміністрування мережі, особливо в середовищах з великою кількістю мобільних клієнтів. Робота протоколу базується на клієнт-серверній архітектурі та використовує транспортний протокол UDP, де обмін повідомленнями відбувається за моделлю DORA (Discover, Offer, Request, Acknowledge). Сервер прослуховує порт 67, а клієнти відправляють запити з порту 68.

Процес розгортання служби DHCP на базі Ubuntu Server розпочинається з оновлення індексу пакетів для гарантування завантаження актуальних версій програмного забезпечення. Встановлення здійснюється за допомогою пакетного менеджера apt шляхом інсталяції пакета isc-dhcp-server. Характерною особливістю першого запуску служби після інсталяції є можлива поява повідомлення про помилку запуску. Це є штатною ситуацією, зумовленою тим, що конфігураційний файл за замовчуванням не містить визначень підмереж, які відповідали б наявним мережевим інтерфейсам сервера. Для коректної роботи служби необхідно виконати налаштування прив’язки до мережевого інтерфейсу. Це здійснюється шляхом редагування файлу /etc/default/isc-dhcp-server, де у параметрі INTERFACESv4 вказується ім’я фізичного інтерфейсу (наприклад, eth0 або ens33), через який сервер буде обслуговувати запити клієнтів. Таке обмеження є важливим заходом безпеки, що запобігає випадковій видачі адрес у зовнішні мережі [36].

Основним конфігураційним файлом сервера є /etc/dhcp/dhcpd.conf. Перед початком налаштування рекомендується створити резервну копію оригінального файлу для можливості відкату змін. Структура файлу складається з глобальних параметрів та секцій оголошення підмереж. На початку файлу визначаються загальні налаштування для всіх клієнтів: доменне ім’я та адреси DNS-серверів.

Важливим параметром є директива authoritative, розкоментування якої вказує, що даний DHCP-сервер є головним у мережі. Якщо цей параметр не активовано, сервер ігноруватиме запити клієнтів на поновлення адрес, виданих іншими серверами, що може призвести до затримок у підключенні. Також задаються параметри часу оренди: default-lease-time (час у секундах, на який видається адреса, якщо клієнт не запитав інше) та max-lease-time (максимально допустимий час оренди).

Ключовим етапом конфігурування DHCP на базі Ubuntu Server є оголошення підмережі. Синтаксис вимагає опису мережі, яка відповідає IP-адресі інтерфейсу сервера. У блоці subnet вказується адреса мережі та маска підмережі. Внутрішні параметри блоку включають діапазон адрес для динамічного розподілу, який визначає пул доступних IP-адрес (наприклад, від 192.168.1.100 до 192.168.1.200). Окрім діапазону, обов’язково вказується шлюз за замовчуванням, який клієнти використовуватимуть для виходу в інші мережі. Важливо пам’ятати, що адреса самого DHCP-сервера повинна бути статичною і знаходитися поза межами діапазону динамічної видачі range, щоб уникнути конфлікту адрес.

Окрім динамічного розподілу, ISC DHCP Server дозволяє налаштовувати резервування адрес, що гарантує отримання конкретним пристроєм незмінної IP-адреси. Це є необхідним для мережевих принтерів, файлових серверів або точок доступу. Резервування реалізується через блок host, у якому вказується довільне ім’я хоста, MAC-адреса мережевої карти пристрою (параметр hardware ethernet) та фіксована IP-адреса, що присвоюється цьому клієнту. Така конфігурація дозволяє централізовано керувати адресацією критичної інфраструктури без необхідності ручного налаштування мережевих параметрів на кожному окремому пристрої.

Після завершення редагування конфігураційних файлів виконується перезапуск служби isc-dhcp-server за допомогою системи ініціалізації systemd. Для діагностики роботи сервера використовується команда перевірки статусу служби, а також аналіз системних журналів. Інформація про видані адреси зберігається у файлі бази даних оренди /var/lib/dhcp/dhcpd.leases. Цей файл містить записи про всі активні та минулі оренди, включаючи час видачі, час закінчення дії, MAC-адресу клієнта та його ім'я хоста. Періодичний моніторинг цього файлу дозволяє адміністратору оцінювати утилізацію адресного простору та виявляти несанкціоновані підключення до локальної мережі. Налаштування DHCP-сервера є фундаментом для подальшого розгортання мережевих служб, зокрема системи доменних імен.

Система доменних імен (DNS) є ієрархічною розподіленою базою даних, що забезпечує перетворення доменних імен, зрозумілих людині, у IP-адреси, необхідні для маршрутизації пакетів. У середовищі ОС Linux адміністрування DNS традиційно передбачає використання програмного забезпечення BIND (Berkeley Internet Name Domain).

Архітектура DNS розрізняє кілька типів серверів залежно від їхньої ролі в обробці запитів та зберіганні даних зон. Основним типом є первинний або майстер-сервер (Primary/Master DNS Server). Цей сервер зберігає оригінальні файли зон, має повноваження на внесення змін до записів і вважається авторитетним джерелом інформації для відповідного домену. Саме на ньому адміністратор створює та редагує записи ресурсів [37].

Для забезпечення відмовостійкості та розподілу навантаження використовуються вторинні або підлеглі сервери (Secondary/Slave DNS Server). Вторинний сервер не містить власних файлів зон, що редагуються вручну; натомість він отримує копію даних зони з первинного сервера через механізм трансферу зони. Вторинний сервер також є авторитетним для зони, але функціонує в режимі «тільки для читання» відносно бази даних доменних імен.

Процес розгортання DNS-сервера на базі Ubuntu Server розпочинається з підготовки системи. Першочерговим кроком є оновлення списків пакетів репозиторіїв командою apt update та оновлення встановленого програмного забезпечення командою apt upgrade. Це гарантує наявність останніх виправлень безпеки, що є критично важливим для служби, яка обробляє зовнішні запити. Наступним етапом виконується встановлення пакета bind9, який містить сам програмний модуль служби DNS, та bind9utils, що надає набір утиліт для діагностики та керування (наприклад, dig, rndc). Конфігураційні файли BIND в Ubuntu за замовчуванням розміщуються в каталозі /etc/bind (рис. 15.8).

```
root@ns1:~# ls -la /etc/bind/
drwxr-sr-x   2 root bind 4096 Sep 22 16:57 .
drwxr-xr-x 111 root root 4096 Sep 21 18:42 ..
-rw-r--r--   1 root root 2403 Jul 16 18:16 bind.keys
-rw-r--r--   1 root root  255 Jan 25  2024 db.0
-rw-r--r--   1 root root  271 Jan 25  2024 db.127
-rw-r--r--   1 root root  237 Jan 25  2024 db.255
-rw-r--r--   1 root root  353 Jan 25  2024 db.empty
-rw-r--r--   1 root root  270 Jan 25  2024 db.local
-rw-r--r--   1 root bind  458 Jan 25  2024 named.conf
-rw-r--r--   1 root bind  498 Jan 25  2024 named.conf.default-zones
-rw-r--r--   1 root bind  165 Jan 25  2024 named.conf.local
-rw-r--r--   1 root bind  846 Jan 25  2024 named.conf.options
-rw-r-----   1 bind bind  100 Sep 21 18:42 rndc.key
-rw-r--r--   1 root root 1317 Jan 25  2024 zones.rfc1918
```

Рисунок 15.8 – Файли конфігурації DNS-сервера [37]

Архітектура конфігурації BIND модульна. Головний файл named.conf зазвичай не редагується напряму, а містить посилання на інші файли, що дозволяє логічно розділити

налаштування. Файл named.conf.options містить глобальні параметри роботи сервера, налаштування безпеки та шляхів. Файл named.conf.local використовується для оголошення зон (прямої та зворотної), які обслуговує даний сервер. Файл named.conf.default-zones описує стандартні зони, такі як localhost та root hints, і зазвичай залишається без змін.

Налаштування глобальних параметрів DNS здійснюється у файлі /etc/bind/named.conf.options. Для підвищення безпеки створюється список контролю доступу (ACL) з назвою local-network, який визначає довірені підмережі (наприклад, 192.168.1.0/24), яким дозволено надсилати рекурсивні запити до сервера. У блоці options задаються ключові директиви: directory вказує робочу папку кешу; dnssec-validation auto активує перевірку підписів DNSSEC; allow-query обмежує коло клієнтів, що можуть опитувати сервер, використовуючи створений ACL; recursion yes дозволяє серверу виконувати запити до зовнішніх доменів від імені клієнтів; forwarders визначає IP-адреси вищестоящих DNS-серверів (наприклад, 8.8.8.8), куди перенаправляються запити, які сервер не може вирішити самостійно. Також налаштовуються параметри прослуховування мережевих інтерфейсів через директиви listen-on (для IPv4) та listen-on-v6 (для IPv6). Після внесення змін служба перезапускається командою systemctl restart bind9 [37].

Наступним кроком є оголошення зон у файлі /etc/bind/named.conf.local. Для прикладу, створюється пряма зона для домену lab.com та зворотна зона для мережі 192.168.1.0/24. У блоці конфігурації кожної зони вказується тип type master, що визначає роль сервера як первинного, та параметр file, який вказує шлях до файлу з записами ресурсів (наприклад, /etc/bind/zones/forward.lab.com.db). Директива allow-update { none; }; забороняє динамічне оновлення зон, що підвищує безпеку статичної конфігурації. Рекомендується створити окремий каталог /etc/bind/zones для зберігання файлів зон та надати відповідні права доступу користувачу bind.

Наповнення файлу прямої зони починається з директиви $TTL та запису SOA (Start of Authority), який описує основні параметри зони: серійний номер, який необхідно збільшувати при кожній зміні, інтервали оновлення, повторної спроби та застарівання. Далі додаються записи NS (Name Server), що вказують на сам DNS-сервер, та записи типу A (Address), які зіставляють імена хостів (ns1, laptop1, pc1) з їхніми IP-адресами. Наприклад, запис ns1 IN A 192.168.1.10 пов’язує ім’я сервера імен з його адресою (рис. 15.9).

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
@       IN      A       127.0.0.1
@       IN      AAAA    ::1
```

Рисунок 15.9 – Файл зони для інтерфейсу зворотного зв’язку [37]

Конфігурація зворотної зони, яка відповідає за перетворення IP-адрес у доменні імена, виконується аналогічно, але замість записів типу A використовуються записи PTR. Файл зворотної зони також містить SOA та NS записи. Записи PTR мають формат, де вказується останній октет IP-адреси та відповідне йому повне доменне ім’я (FQDN). Наприклад, запис 10 IN PTR ns1.lab.com. дозволяє визначити ім’я хоста за IP-адресою 192.168.1.10 (рис. 15.10). Якщо в мережі не використовується протокол IPv6, рекомендується примусово перевести BIND у режим роботи тільки з IPv4, відредагувавши файл /etc/default/named та додавши параметр -4 до змінної OPTIONS [37].

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
1.0.0   IN      PTR     localhost.
```

Рисунок 15.10 – Зворотна зона для інтерфейсу зворотного зв'язку [37]

Перевірка коректності налаштувань є обов'язковим етапом перед введенням сервера в експлуатацію. Для цього використовуються утиліти синтаксичного контролю: named-checkconf для перевірки головних конфігураційних файлів та named-checkzone для перевірки файлів зон. Якщо помилок не виявлено, виконується перезапуск служби. Тестування працездатності здійснюється з клієнтських машин за допомогою утиліт ping (перевірка вирішення імен), а також спеціалізованих інструментів nslookup, dig або host. Успішне вирішення прямих (ім'я в IP) та зворотних (IP в ім'я) запитів свідчить про коректну роботу первинного DNS-сервера [37].

Для забезпечення відмовостійкості інфраструктури виконується налаштування вторинного DNS-сервера. Цей процес вимагає попередніх змін на первинному сервері. У файлі зони майстер-сервера необхідно додати запис NS та A для вторинного сервера (наприклад, ns2) (рис. 15.11). Крім того, у файл named.conf.local на майстер-сервері додається директива allow-transfer, в якій вказується IP-адреса вторинного сервера (рис. 15.12). Це дозволяє передачу даних зони виключно авторизованому вторинному серверу. На стороні вторинного сервера процес встановлення програмного забезпечення ідентичний первинному. Відмінності полягають у конфігурації файлу /etc/bind/named.conf.local. При оголошенні зон на вторинному сервері використовується тип «type slave».

```
vi /etc/bind/zones/forward.lab.com.db

$TTL    604800
@       IN      SOA     ns1.lab.com. root.lab.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

@       IN      NS      ns1.lab.com.
@       IN      NS      ns2.lab.com.

ns1     IN      A     192.168.1.10
ns2     IN      A     192.168.1.11

laptop1 IN      A     192.168.1.21
laptop2 IN      A     192.168.1.22
pc1     IN      A     192.168.1.23
pc2     IN      A     192.168.1.24
pc3     IN      A     192.168.1.25
phone   IN      A     192.168.1.26
printer IN      A     192.168.1.27

:x //save the file
```

Рисунок 15.11 – Записи вторинного DNS-сервера на головному DNS-сервері [37]

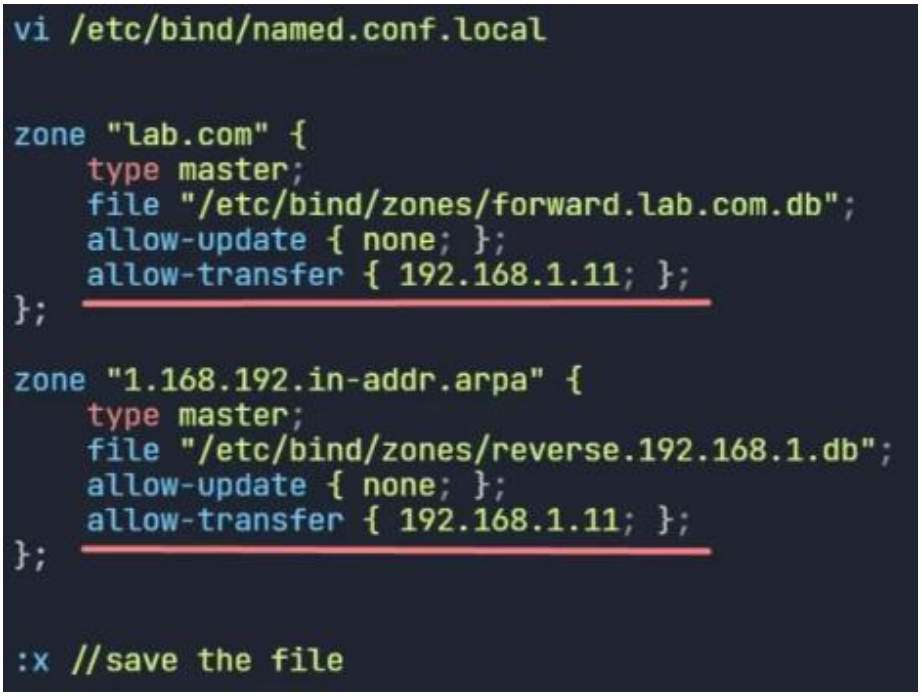

Рисунок 15.12 – Надання дозволу на вторинний DNS-сервер для передачі зони [37]

Обов'язково додається директива masters, яка містить IP-адресу первинного сервера (наприклад, 192.168.1.10), звідки будуть завантажуватися дані. Параметр file вказує локальний шлях, куди вторинний сервер збереже отриману копію зони. Після перезапуску служби на вторинному сервері ініціюється процес трансферу зони, і файли зон автоматично створюються в зазначеному каталозі, що можна перевірити переглядом вмісту директорії або аналізом системних журналів. Це завершує побудову системи доменних імен на базі Linux Server.

Механізми віддаленого доступу (SSH)

У сучасних мережевих інфраструктурах забезпечення безпечного віддаленого адміністрування серверних вузлів є важливою вимогою, яка реалізується за допомогою протоколу прикладного рівня SSH (Secure Shell).

Протокол SSH був розроблений як захищена альтернатива застарілим і незахищеним інструментам, таким як Telnet, rlogin та RSH, які передавали автентифікаційні дані та командний трафік у відкритому вигляді. Архітектура SSH базується на клієнт-серверній моделі, де серверний компонент (sshd) очікує на вхідні з'єднання на певному TCP-порту (стандартно 22), а клієнтське програмне забезпечення (ssh) ініціює з'єднання. Фундаментальною особливістю протоколу є обов'язкове шифрування всього трафіку, включаючи процедуру автентифікації, що досягається використанням гібридної криптографії. Асиметричні алгоритми застосовуються для обміну ключами сесії та автентифікації сторін, тоді як симетричні алгоритми використовуються для швидкого шифрування потоку даних в рамках встановленої сесії.

Процес розгортання серверної частини SSH у середовищі Ubuntu Server є стандартизованою процедурою, що виконується через систему керування пакетами. Для встановлення SSH-сервера в терміналі застосовуються команди оновлення індексу пакетів «sudo apt update» та безпосередньої інсталяції пакунка «sudo apt install openssh-server –y», як це показано на рисунку 15.13. Виконання цих команд ініціює завантаження бінарних файлів, генерацію унікальних хост-ключів сервера, які слугують його цифровим відбитком для запобігання атакам типу «людина посередині», та автоматичну реєстрацію служби в системі ініціалізації systemd. Важливо зазначити, що хост-ключі генеруються один раз при встановленні, і їх зміна призведе до появи попереджень про порушення безпеки у клієнтів, що раніше підключалися до цього вузла.

```
administrator@adminserv:~$ sudo apt install openssh-server -y
Наступні пакунки були встановлені автоматично і більше не потрібні:
  linux-headers-6.14.0-15          linux-modules-extra-6.14.0-15-generic
  linux-headers-6.14.0-15-generic  linux-tools-6.14.0-15
  linux-modules-6.14.0-15-generic  linux-tools-6.14.0-15-generic
Використовуйте 'sudo apt autoremove' щоб видалити їх.

Installing:
  openssh-server
```

Рисунок 15.13 – Встановлення SSH-сервера [36]

Після завершення процесу інсталяції, важливим етапом є верифікація коректності запуску служби. Для цього перевіряється робота служби SSH-сервера, використовуючи команду «sudo systemctl status ssh» в терміналі, яка повинна повернути статус «active (running)». Окрім статусу процесу, необхідно впевнитися у коректності прив'язки до мережевих сокетів. Також після цього здійснюється перевірка, чи «слухає» сервер порт 22 командою «ss -tlnp | grep 22» (рис. 15.14). У виводі цієї команди прапорці «LISTEN» свідчать про готовність сервера приймати вхідні з'єднання, а відображення адреси «0.0.0.0» або «*» вказує на прослуховування всіх доступних мережевих інтерфейсів.

```
administrator@adminserv:~$ sudo systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: inactive (dead)
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
administrator@adminserv:~$
administrator@adminserv:~$
administrator@adminserv:~$ ss -tlnp | grep 22
LISTEN 0      4096         0.0.0.0:22        0.0.0.0:*
LISTEN 0      4096            [::]:22           [::]:*
administrator@adminserv:~$
```

Рисунок 15.14 – Перевірка роботи SSH-сервера [36]

Конфігурування параметрів роботи служби здійснюється шляхом редагування текстових файлів налаштувань. Важливо розрізняти файл налаштувань клієнта (ssh_config) та файл налаштувань сервера (sshd_config). Для зміни налаштувань SSH-сервера слід відкрити конфігураційний файл, що міститься за адресою «/etc/ssh/sshd_config». Там, знайшовши потрібне поле, необхідно провести зміни згідно з політикою безпеки. До типових налаштувань належить зміна стандартного порту для зменшення кількості автоматизованих атак, заборона прямого входу суперкористувача (параметр PermitRootLogin no) та обмеження списку дозволених користувачів (AllowUsers). Після будь-якої зміни конфігурації виконується перезапуск SSH-сервера командою «sudo systemctl restart ssh», що змушує перечитати файл налаштувань без розриву вже встановлених активних з'єднань [36].

Забезпечення мережевої доступності сервісу вимагає коректного налаштування міжмережевого екрану сервера. Для подальшого правильного функціонування SSH-сервера проводяться налаштування відповідних параметрів брандмауера UFW (Uncomplicated Firewall), який є стандартним інструментом в Ubuntu. Для цього в терміналі вводиться команда «sudo ufw allow 22/tcp», яка створює правило дозволу вхідного трафіку на порт 22 по протоколу TCP. Після цього застосовується команда «sudo ufw enable» для активації брандмауера та завантаження правил (рис. 15.15). Коли ці дії зроблено, то SSH-сервер має коректно працювати і бути доступним із зовнішньої мережі, при цьому решта портів залишатимуться закритими згідно з політикою за замовчуванням.

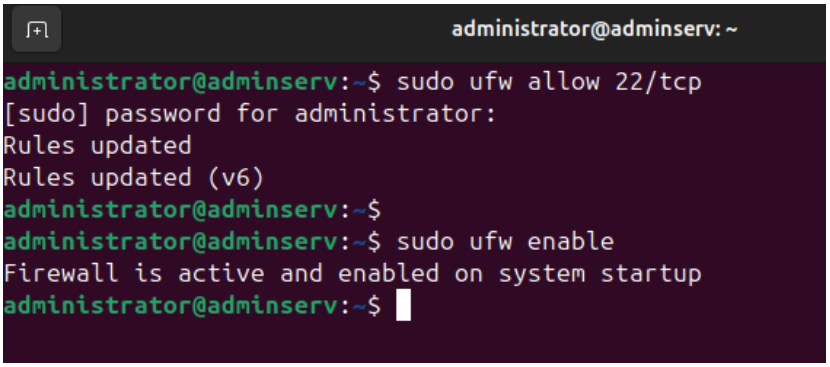

Рисунок 15.15 – Налаштування брандмауера для роботи SSH-сервера [36]

Найбільш надійним методом автентифікації в SSH вважається використання пари криптографічних ключів замість паролів. Цей метод базується на асиметричній криптографії: користувач генерує пару ключів (публічний та приватний) на локальній машині. Публічний ключ, який не є секретним, копіюється на сервер у файл ~/.ssh/authorized_keys у домашньому каталозі користувача. При спробі з'єднання сервер шифрує випадкове повідомлення за допомогою публічного ключа користувача. Клієнт може розшифрувати це повідомлення і підтвердити свою особу тільки за наявності відповідного приватного ключа. Цей підхід нівелює ризики, пов'язані зі слабкими паролями та брутфорс-атаками [36].

Окрім надання віддаленого термінального доступу, протокол SSH забезпечує захищений транспорт для інших протоколів та утиліт. Зокрема, утиліта SCP (Secure Copy) та протокол SFTP (SSH File Transfer Protocol) використовують встановлений SSH-канал для шифрованої передачі файлів між хостами, замінюючи незахищений FTP. Також SSH підтримує механізм тунелювання, який дозволяє інкапсулювати трафік інших застосунків (наприклад, запити до бази даних або веб-трафік) всередину зашифрованої сесії SSH. Це дозволяє адміністраторам отримувати доступ до внутрішніх сервісів локальної мережі сервера, які не мають прямого виходу в Інтернет, створюючи своєрідний ситуативний VPN-канал для безпечного адміністрування складної інфраструктури.

Контроль процесів і логів

У операційній системі Ubuntu Server, як і в будь-якій Unix-подібній системі, поняття процесу є фундаментальним для розуміння принципів функціонування програмного забезпечення. Процес визначається як екземпляр програми, що знаходиться в стані виконання, якому ядро операційної системи виділяє певний адресний простір у пам'яті та набір системних ресурсів. Кожен процес у системі ідентифікується унікальним цілим числом, відомим як ідентифікатор процесу (Process Identifier – PID). Ієрархія процесів будується деревоподібно: кожен новий процес породжується батьківським процесом (Parent Process) за допомогою системного виклику fork(), отримуючи при цьому ідентифікатор батька (PPID). Винятком є лише процес ініціалізації systemd (у сучасних версіях Ubuntu), який має PID 1 і запускається безпосередньо ядром під час завантаження системи. Саме systemd виступає коренем дерева процесів користувацького простору і відповідає за запуск решти системних служб та терміналів.

Керування процесами неможливе без розуміння їхніх станів, оскільки процес не завжди активно використовує центральний процесор. У планувальнику завдань Linux (Completely Fair Scheduler – CFS) розрізняють декілька основних станів. Стан «Running» (R) означає, що процес або виконується на процесорі в даний момент, або знаходиться в черзі на виконання. Стан «Sleeping» поділяється на перериваний (S – Interruptible sleep) та неперериваний (D – Uninterruptible sleep). Перериваний сон характерний для процесів, що очікують на певну подію (наприклад, введення даних користувачем), і можуть бути розбуджені сигналом. Неперериваний сон зазвичай виникає при очікуванні операцій введення-виведення на апаратному рівні (дискові операції), і такий процес неможливо примусово завершити до закінчення операції. Особливої уваги адміністратора вимагають процеси у стані «Zombie» (Z) – це процеси, які завершили своє виконання, звільнили ресурси, але запис про них все ще залишається в таблиці процесів ядра, оскільки батьківський процес не зчитав код їх завершення через виклик wait() [36].

Для отримання інформації про поточний стан системи в Ubuntu Server використовується віртуальна файлова система /proc. Ця директорія не містить реальних файлів на жорсткому диску, а є інтерфейсом до структур даних ядра. Кожен запущений процес має власну піддиректорію в /proc, назва якої відповідає його PID (наприклад, /proc/1234/). У цій директорії містяться файли з повною інформацією про процес: cmdline (команда запуску), environ (змінні оточення), fd (відкриті файлові дескриптори), status (поточний стан та використання пам'яті). Інструменти моніторингу, такі як ps, top або htop, фактично зчитують та форматують інформацію саме з цієї файлової системи, надаючи адміністратору зрозуміле візуальне представлення.

Базовим інструментом для отримання миттєвого зрізу активності процесів є утиліта ps (process status). Для повноцінного аналізу в серверному середовищі найчастіше застосовується комбінація ключів aux (стандарт BSD) або -ef (стандарт System V). Виконання команди ps aux виводить таблицю, де відображаються користувач-власник процесу, PID, відсоток використання процесора та пам'яті віртуальна та резидентна пам'ять, статус, час старту та сама команда. Важливим аспектом аналізу є розуміння параметра TTY: якщо в цьому полі стоїть знак питання ?, це свідчить про те, що процес є службою, яка не прив'язана до жодного термінала. Це типова поведінка для веб-серверів (Apache, Nginx) або баз даних, які працюють у фоновому режимі.

Для спостереження за динамікою споживання ресурсів у реальному часі використовуються інтерактивні утиліти top та його вдосконалений аналог htop. Утиліта top дозволяє сортувати процеси за споживанням ЦП або ОЗП, а також надає загальну статистику системи: час роботи (uptime), кількість користувачів та середнє навантаження (Load Average). Показник Load Average, що відображається як три числа (за 1, 5 та 15 хвилин), є критичною метрикою для оцінки продуктивності сервера. Він показує середню кількість процесів, які знаходяться в стані виконання або в черзі на виконання (включаючи ті, що перебувають у стані непереривного сну). Якщо значення Load Average перевищує кількість доступних ядер процесора, це свідчить про перевантаження системи та виникнення черг, що потребує втручання адміністратора для оптимізації або масштабування ресурсів [36].

Механізм керування процесами базується на використанні сигналів – це засіб міжпроцесної взаємодії, що дозволяє ядру або одному процесу передати керуючу команду іншому. Для надсилання сигналів використовується команда kill. Попри свою назву, ця утиліта призначена не лише для знищення процесів, а й для керування ними. За замовчуванням kill надсилає сигнал SIGTERM (код 15), який просить процес коректно завершити роботу, закрити відкриті файли та звільнити пам'ять. Якщо процес завис і не реагує на SIGTERM, застосовується сигнал SIGKILL (код 9), який обробляється безпосередньо ядром і миттєво припиняє виконання процесу без можливості збереження даних. Ще одним важливим сигналом є SIGHUP (код 1), який використовується для перезавантаження конфігурації служб без повної зупинки процесу.

У багатокористувацькому середовищі Linux важливим є керування пріоритетами виконання завдань. Кожен процес має атрибут «niceness» (значення nice), який визначає його відношення до інших процесів у боротьбі за процесорний час. Діапазон значень варіюється від -20 (найвищий пріоритет) до +19 (найнижчий пріоритет). За замовчуванням процеси запускаються з пріоритетом 0. Звичайний користувач може лише знижувати пріоритет своїх процесів (збільшувати значення nice), тоді як суперкористувач (root) має право підвищувати пріоритет (зменшувати значення nice). Для запуску процесу з нестандартним пріоритетом використовується команда nice, а для зміни пріоритету вже запущеного процесу – renice. Це дозволяє адміністратору, наприклад, надати вищий пріоритет процесу бази даних і знизити пріоритет для завдань резервного копіювання, що виконуються у фоні.

Керування фоновим та активним виконанням процесів здійснюється засобами командної оболонки Bash. Якщо виконання команди займає тривалий час і блокує термінал, її можна запустити у фоновому режимі, додавши символ амперсанда & в кінці рядка. Вже запущений процес можна призупинити комбінацією клавіш Ctrl+Z, що надсилає сигнал SIGSTOP, а потім перевести у фоновий режим командою bg або повернути в активний режим командою fg. Для перегляду списку завдань поточної сесії оболонки використовується

команда jobs. Однак, процеси, запущені в терміналі, є нащадками процесу оболонки, тому при закритті сесії SSH вони отримають сигнал SIGHUP і завершаться. Щоб уникнути цього, використовують утиліту nohup або термінальні мультиплексори (screen, tmux), які дозволяють процесам продовжувати роботу після від'єднання користувача.

В контексті Ubuntu Server основним інструментом керування серверними процесами (службами) є система systemd, яка оперує поняттям «юнітів». Контроль процесів здійснюється через утиліту systemctl. На відміну від прямого запуску бінарних файлів, systemd забезпечує стандартизоване оточення, автоматичний перезапуск у разі збоїв, керування залежностями та логування. Статус служби перевіряється командою systemctl status <service_name>, яка надає детальну інформацію про PID головного процесу, дерево дочірніх процесів, споживання пам'яті та останні рядки логів. Керування станом здійснюється командами start, stop, restart та reload. Важливою функцією є systemctl enable/disable, яка керує символічними посиланнями для автоматичного запуску служби під час завантаження системи [36].

Оперативний контроль логів процесів, стосується роботи зі стандартними потоками введення-виведення (stdout та stderr) у реальному часі. У традиційній моделі Unix процеси виводять інформаційні повідомлення у стандартний потік виведення (stdout), а повідомлення про помилки – у стандартний потік помилок (stderr). Адміністратор повинен вміти перехоплювати та аналізувати ці потоки безпосередньо під час виконання. Для цього використовуються оператори перенаправлення. Наприклад, конструкція command > log.txt 2>&1 перенаправляє обидва потоки в один файл, що дозволяє зберегти повну історію виконання. Для відкидання непотрібного виводу використовується спеціальний пристрій /dev/null, який діє як «чорна діра» для даних.

Для спостереження за логами процесів у режимі реального часу, особливо тих, що записують дані у текстові файли, незамінною є утиліта tail з ключем -f (follow). Виконання команди tail -f /var/log/syslog або tail -f /var/log/nginx/access.log дозволяє адміністратору бачити нові рядки журналу в момент їх появи. Це є основним методом діагностики при налагодженні роботи сервісів, коли необхідно співставити дії користувача (наприклад, HTTP-запит) з реакцією сервера. У середовищі systemd аналогічний функціонал надає команда journalctl -f -u <service_name>, яка підключається до системного журналу і виводить повідомлення конкретної служби в міру їх надходження, дозволяючи відстежувати поведінку демонів, що не мають власних лог-файлів.

Крім того, контроль процесів включає моніторинг використання ними файлових дескрипторів та мережевих сокетів. У ОС Linux «все є файлом», тому мережеве з'єднання для процесу виглядає як відкритий файл. Утиліта lsof (List Open Files) дозволяє визначити, які файли відкриті конкретним процесом (ключ -p PID), або навпаки – які процеси використовують конкретний файл або порт (ключ -i). Це критично важливо для діагностики помилок «File in use» або виявлення несанкціонованої мережевої активності. Наприклад, команда lsof -i :80 покаже всі процеси, що взаємодіють через порт 80. Такий глибокий аналіз взаємозв'язків між процесами та ресурсами файлової системи є завершальним етапом повного контролю над станом сервера перед переходом до налаштування систем довгострокового зберігання та аналізу журналів.

Налаштування Apache/Nginx

У сучасній інженерії веб-систем та адмініструванні серверних інфраструктур одним із ключових завдань є забезпечення надійної, безпечної та продуктивної доставки контенту кінцевому користувачеві. Реалізація цього завдання часто вимагає виходу за межі використання єдиного монолітного веб-сервера. Найбільш ефективним підходом, що зарекомендував себе в індустрії, є побудова багаторівневої архітектури, де поєднуються сильні сторони різних програмних рішень.

У цьому контексті розглядається методологія розгортання та конфігурації гібридної системи, що складається з двох найпопулярніших веб-серверів: Apache2 та Nginx. У запропонованій архітектурі Nginx виконуватиме роль зворотного проксі-сервера, що приймає вхідні запити та здійснює їх попередню обробку, тоді як Apache2 функціонуватиме як сервер додатків, безпосередньо обробляючи логіку веб-ресурсу [38].

Процес побудови веб-серверної інфраструктури розпочинається з ретельного підготовчого етапу, який визначає фундамент стабільності майбутньої системи. Оскільки веб-сервери Nginx та Apache2 є нативними для Unix-подібних систем, першочерговою вимогою є наявність налаштованого сервера під управлінням операційної системи Linux. У контексті даного навчального матеріалу базовою платформою обрано дистрибутив Ubuntu, який базується на архітектурі Debian і широко використовується в корпоративному секторі завдяки своїй стабільності та широкій підтримці спільноти. Слід зазначити, що для користувачів, які працюють у середовищі Windows, існує можливість емуляції необхідного оточення за допомогою підсистеми WSL (Windows Subsystem for Linux), що дозволяє використовувати інструментарій Linux без необхідності повної переінсталяції операційної системи або використання важковагових віртуальних машин. Критично важливими передумовами для успішної реалізації проекту є наявність стабільного підключення до мережі Інтернет для завантаження пакетів із репозиторіїв, а також наявність привілеїв суперкористувача (root) або прав на виконання команд через механізм sudo, оскільки конфігурація мережевих служб та модифікація системних файлів вимагають підвищеного рівня доступу [38].

Наступним кроком підготовчої фази є забезпечення наявності веб-додатку, який буде обслуговуватися розгорнутою інфраструктурою. Адміністратор системи постає перед вибором: розробити власний програмний продукт, що надає повний контроль над архітектурою коду та функціональними можливостями, або використати готове рішення для пришвидшення процесу розгортання та фокусування на налаштуванні серверної частини.

У навчальних цілях, для стандартизації процесу та уникнення помилок на етапі кодування, рекомендується використання попередньо підготовленого веб-додатку, доступного у відповідному репозиторії на платформі GitHub. Після завантаження вихідного коду на локальну машину або сервер, необхідно провести первинну верифікацію цілісності файлів. Це виконується шляхом відкриття головного файлу index.html у локальному веб-браузері. Відображення коректного графічного інтерфейсу свідчить про готовність додатку до розгортання, проте на цьому етапі він функціонує лише як набір локальних файлів і ще не інтегрований у серверну інфраструктуру, що підводить до необхідності інсталяції відповідного серверного програмного забезпечення.

Першим компонентом серверної тріади, що підлягає інсталяції, є веб-сервер Apache2. Цей сервер вирізняється своєю модульністю, надійністю та підтримкою широкого спектра технологій обробки динамічного контенту. Процедура інсталяції в середовищі Ubuntu виконується через термінальний інтерфейс, який є основним інструментом системного адміністратора. Перед початком інсталяції критично важливо актуалізувати інформацію про доступні пакети в системних репозиторіях. Виконання команди sudo apt-get update ініціює з'єднання з серверами оновлень та завантаження актуальних списків пакетів, що гарантує встановлення останніх стабільних версій програмного забезпечення та наявність необхідних патчів безпеки. Після оновлення індексу виконується безпосередня інсталяція пакету Apache2 за допомогою команди sudo apt-get install apache2 (рис. 15.16). Пакетний менеджер автоматично вирішує залежності, завантажує бінарні файли та конфігураційні скрипти, розміщуючи їх у відповідних директоріях файлової системи.

```
administrator@adminserv:~$ sudo apt update
[sudo] password for administrator:
В кеші:1 http://ua.archive.ubuntu.com/ubuntu plucky InRelease
В кеші:2 http://ua.archive.ubuntu.com/ubuntu plucky-updates InRelease
В кеші:3 http://ua.archive.ubuntu.com/ubuntu plucky-backports InRelease
В кеші:4 http://security.ubuntu.com/ubuntu plucky-security InRelease
18 packages can be upgraded. Run 'apt list --upgradable' to see them.
administrator@adminserv:~$ sudo apt install apache2 -y
Наступні пакунки були встановлені автоматично і більше не потрібні:
  linux-headers-6.14.0-15          linux-modules-extra-6.14.0-15-generic
  linux-headers-6.14.0-15-generic  linux-tools-6.14.0-15
  linux-modules-6.14.0-15-generic  linux-tools-6.14.0-15-generic
Використовуйте 'sudo apt autoremove' щоб видалити їх.

Installing:
  apache2
```

Рисунок 15.16 – Встановлення Apache2 [38]

Після завершення процесу розпакування та налаштування пакетів, службу Apache2 необхідно запустити та перевести в активний стан. Керування службами в сучасних дистрибутивах Linux здійснюється системою ініціалізації systemd. Виконання команди sudo systemctl start apache2 ініціює запуск головного процесу веб-сервера. Для перевірки успішності запуску та коректності прив'язки до мережевих інтерфейсів використовується метод тестового запиту [38].

Адміністратор повинен відкрити веб-браузер та ввести в адресний рядок IP-адресу сервера або стандартне доменне ім'я локального хоста – localhost. Якщо процес інсталяції пройшов без помилок, браузер відобразить стандартну сторінку привітання «Apache2 Default Page» (рис. 15.17). Ця сторінка є важливим діагностичним індикатором, який підтверджує, що веб-сервер коректно встановлений, запущений, прослуховує порт 80 за протоколом TCP і має доступ до файлової системи для читання веб-документів.

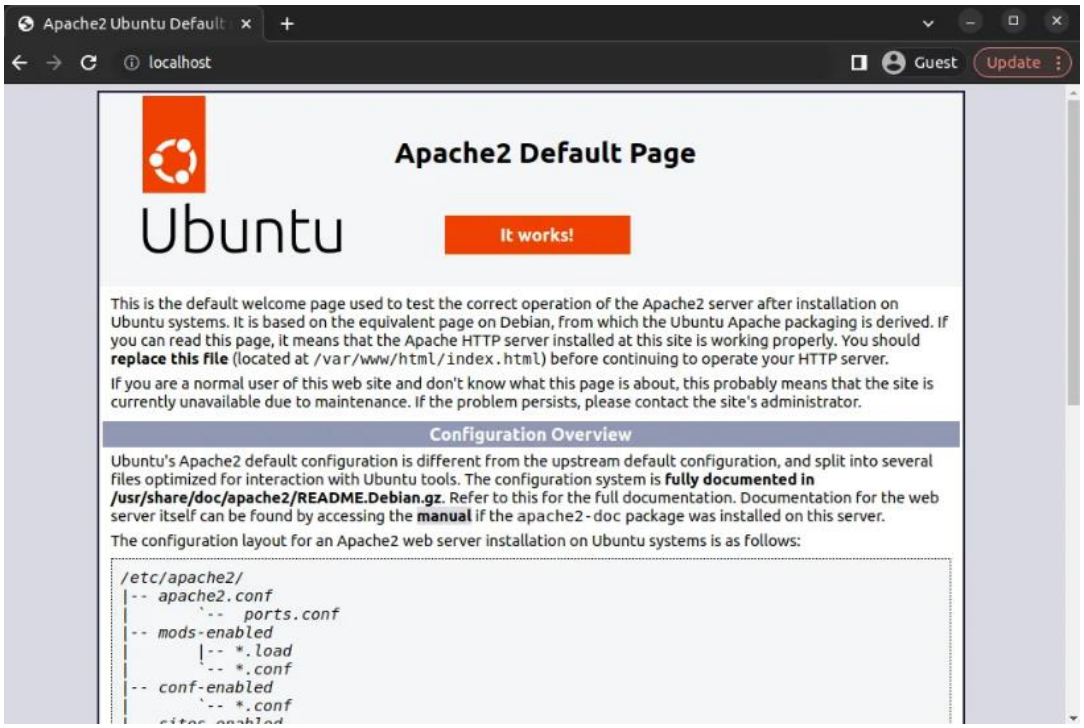

Рисунок 15.17 – Сторінка привітання «Apache2 Default Page» [38]

Інтеграція користувацького веб-додатку в середовище Apache2 вимагає заміни стандартного контенту, що постачається разом із сервером, на файли цільового проекту. Стандартна конфігурація Apache в Ubuntu визначає директорію /var/www/html як кореневу папку документів. Оскільки файл привітання зазвичай має назву index.html, його наявність конфліктуватиме з індексним файлом веб-додатку. Тому, використовуючи термінал, необхідно виконати очищення цільової директорії. Команда sudo rm -r /var/www/html/index.html виконує видалення стандартного файлу. Використання прапора -r (recursive) у даному контексті є запобіжним заходом для гарантованого видалення об'єкта, навіть якщо це директорія, хоча для окремого файлу це не є обов'язковим. Наступним кроком здійснюється перенесення файлів веб-додатку з директорії завантаження до кореневої директорії сервера. Команда sudo cp -r * /var/www/html виконує рекурсивне копіювання всього вмісту поточного каталогу в системну директорію веб-сервера [38].

Після заміни файлів необхідно провести повторне тестування для підтвердження коректності розгортання додатку. Звернення до адреси localhost у веб-браузері повинно призвести до відображення інтерфейсу користувацького веб-додатку замість стандартної сторінки Apache (рис. 15.18). Успішне відображення контенту свідчить про те, що Apache2 налаштований правильно, має необхідні права доступу до нових файлів і готовий виконувати функції бекенд-сервера. Однак, на даному етапі архітектура системи ще не є завершеною, оскільки додаток обслуговується безпосередньо сервером Apache, без проміжного шару проксіювання, який необхідний для забезпечення масштабованості та безпеки згідно з

поставленим технічним завданням. Наступні етапи передбачають розгортання Nginx та налаштування взаємодії між двома серверами.

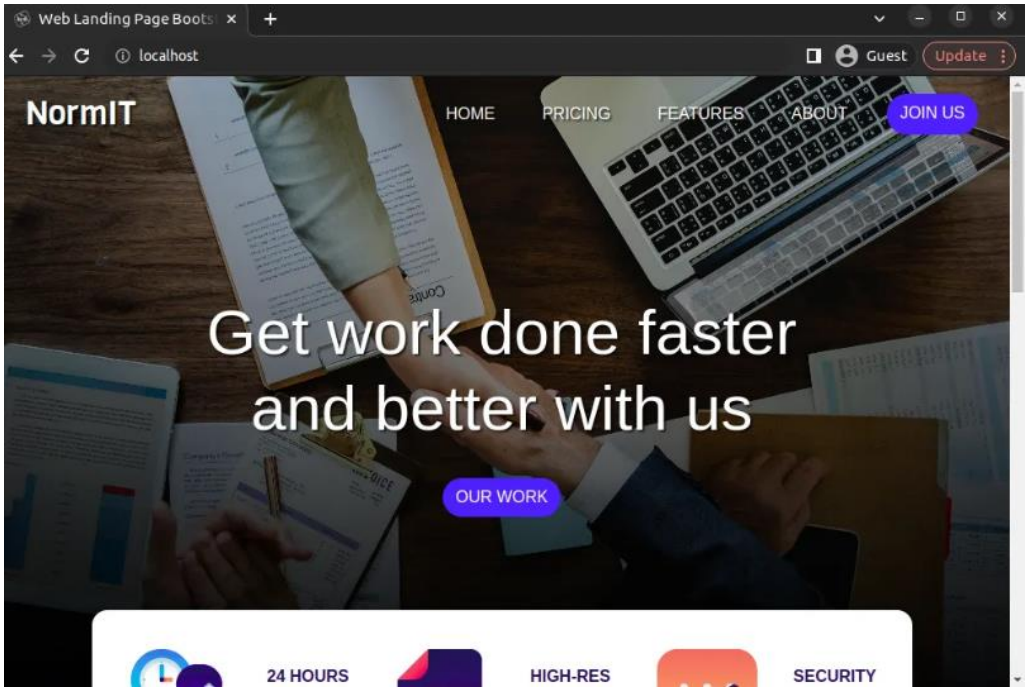

Рисунок 15.18 – Успішне відображення інтерфейсу користувацького веб-додатку [38]

Встановлення веб-сервера Nginx здійснюється за аналогічним алгоритмом, що й встановлення Apache2, використовуючи пакетний менеджер apt. Після оновлення репозиторіїв командою sudo apt-get update, виконується команда sudo apt-get install nginx (рис. 15.19). Ця дія встановлює сервер Nginx у систему.

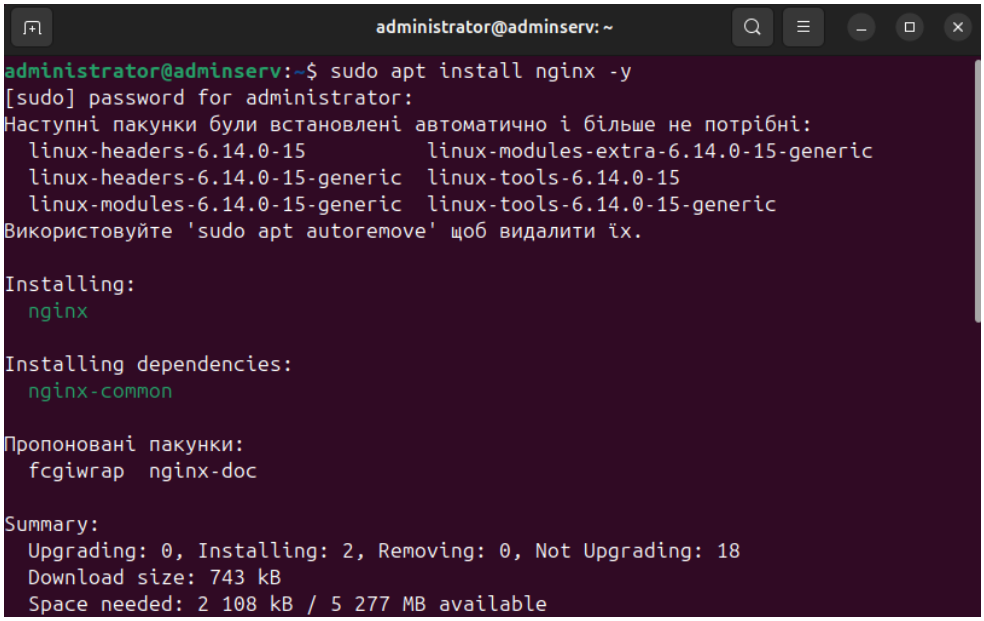

Рисунок 15.19 – Встановлення Nginx [38]

Однак, виникає архітектурна колізія, адже за замовчуванням обидва сервери – і Apache2, і Nginx – налаштовані на прослуховування стандартного HTTP-порту 80. Оскільки два процеси не можуть одночасно використовувати один і той самий мережевий порт на одному IP-інтерфейсі, необхідно змінити конфігурацію Nginx перед його запуском у робочий режим. Стратегія полягає в тому, щоб перенести Nginx на альтернативний порт (порт 3000) для початкового налаштування, а згодом налаштувати його як точку входу.

Для зміни порту прослуховування Nginx необхідно внести правки до головного файлу

конфігурації віртуального хоста. Використовуючи текстовий редактор, наприклад nano, адміністратор відкриває файл за шляхом /etc/nginx/sites-available/default командою sudo nano /etc/nginx/sites-available/default. У структурі файлу необхідно знайти директиви listen, які відповідають за прив'язку до портів. Стандартні записи listen 80 default_server; (для IPv4) та listen [::]:80 default_server; (для IPv6) необхідно модифікувати, замінивши значення 80 на 3000. Таким чином, конфігурація набуде вигляду listen 3000 default_server; та listen [::]:3000 default_server;. Крім того, рекомендується скоригувати директиву index, залишивши в пріоритеті файл index.nginx-debian.html для перевірки роботи самого Nginx, або адаптувати її під потреби додатку. Збереження внесених змін виконується стандартними засобами редактора (CTRL+X, Y, Enter) [38].

Після модифікації конфігураційного файлу необхідно перевірити працездатність Nginx на новому порту. Для цього у веб-браузері вводиться адреса сервера з явним зазначенням порту: localhost:3000. Відображення стандартної сторінки привітання «Welcome to nginx!» підтверджує, що сервер успішно інстальовано, конфігурацію зчитано коректно, і конфлікт портів з Apache2 (який залишається на порту 80) вирішено (рис. 15.20).

Рисунок 15.20 – Відображення стандартної сторінки привітання «Welcome to nginx!» [38]

На цьому етапі в системі паралельно функціонують два веб-сервери, які працюють незалежно один від одного. Наступним, критично важливим кроком, є об'єднання їх у єдину логічну структуру шляхом налаштування Nginx як зворотного проксі-сервера, що дозволить йому приймати запити від клієнтів і перенаправляти їх на бекенд-сервер Apache.

Концепція зворотного проксі є центральною в сучасних веб-архітектурах. Зворотний проксі діє як посередник, що стоїть між клієнтом (браузером користувача) та внутрішнім сервером ресурсів (Apache2). На відміну від прямого проксі, який обслуговує вихідні запити клієнтів до Інтернету, зворотний проксі обслуговує вхідні запити з Інтернету до внутрішніх серверів. Використання такої схеми надає низку вагомих переваг. По-перше, це підвищення рівня безпеки, оскільки зворотний проксі приховує топологію внутрішньої мережі та характеристики бекенд-серверів від зовнішнього світу, виступаючи своєрідним щитом. По-друге, це покращення продуктивності за рахунок можливостей кешування статичного контенту, стиснення даних та ефективного управління з'єднаннями. По-третє, це можливість гнучкого перенаправлення трафіку та балансування навантаження між декількома серверами додатків [38].

Реалізація функціоналу зворотного проксі в Nginx вимагає подальшого редагування файлу конфігурації /etc/nginx/sites-available/default. Ключові зміни вносяться в блок location /, який відповідає за обробку запитів до кореневої директорії сайту. У цьому блоці необхідно додати директиву proxy_pass, яка вказує адресу призначення для перенаправлення запитів. У даному сценарії це буде http://localhost:80, тобто адреса локального сервера Apache. Однак простого перенаправлення недостатньо для коректної роботи сучасних веб-додатків. Необхідно також налаштувати передачу HTTP-заголовків, щоб зберегти контекст запиту. Директива proxy_set_header Host $host; передає оригінальне ім'я хоста, що дозволяє Apache коректно визначати віртуальний хост. Заголовки Upgrade та Connection налаштовуються для підтримки протоколу WebSocket та механізму оновлення з'єднання (HTTP Upgrade mechanism).

Повна конфігурація блоку location має виглядати наступним чином: спочатку вказується proxy_pass http://localhost:80;, потім встановлюється версія протоколу

proxy_http_version 1.1;. Далі слідують директиви встановлення заголовків: proxy_set_header Upgrade $http_upgrade;, proxy_set_header Connection 'upgrade';, proxy_set_header Host $host;. Також додається директива proxy_cache_bypass $http_upgrade;, яка дозволяє оминати кеш для певних типів запитів. Після внесення цих налаштувань та збереження файлу, службу Nginx необхідно перезавантажити командою sudo systemctl restart nginx для застосування змін. Тестування проводиться шляхом звернення до localhost:3000. У разі успішного налаштування, за цією адресою повинен відображатися веб-додаток, який фізично розміщений на Apache, що підтверджує прозоре проксіювання трафіку через Nginx [38].

Кінцевим етапом налаштування є впровадження механізмів захисту, зокрема обмеження частоти запитів. Ця практика є критично важливою для забезпечення стабільності роботи сервера в умовах високого навантаження або зловмисних атак. Обмеження частоти запитів дозволяє контролювати кількість звернень, які сервер приймає від одного клієнта за одиницю часу. Це ефективний інструмент протидії атакам типу DDoS (розподілена відмова в обслуговуванні) та брутфорс-атакам (підбір паролів). Крім того, це дозволяє справедливо розподіляти системні ресурси (процесорний час, оперативну пам’ять) між користувачами, запобігаючи ситуації, коли один клієнт монополізує всі потужності сервера.

Налаштування обмеження частоти запитів в Nginx здійснюється у два етапи. Перший етап – визначення зони обмеження – виконується в основному контексті конфігурації (зазвичай на рівні http, але в спрощеному варіанті можна додати на початку файлу конфігурації сайту поза блоком server). Директива limit_req_zone $binary_remote_addr zone=one:10m rate=30r/m; створює зону з назвою «one». Параметр $binary_remote_addr означає, що ідентифікація клієнтів відбувається за їхньою IP-адресою у бінарному форматі (що економить пам’ять). Розмір зони 10m (10 мегабайт) дозволяє зберігати інформацію про стани десятків тисяч сесій. Параметр rate=30r/m встановлює ліміт швидкості обробки – 30 запитів на хвилину, що еквівалентно одному запиту кожні 2 секунди.

Другий етап – застосування обмеження до конкретного контексту. У блоці location / (або в блоці server) додається директива limit_req zone=one;. Це вказує Nginx використовувати раніше визначену зону «one» для перевірки частоти запитів до даного ресурсу. Після збереження конфігурації та перезапуску служби Nginx (sudo systemctl restart nginx), система починає відстежувати активність клієнтів. Якщо частота запитів від однієї IP-адреси перевищить встановлений поріг (у даному прикладі – частіше ніж раз на 2 секунди), Nginx тимчасово заблокує обробку нових запитів від цього клієнта і поверне HTTP-відповідь з кодом помилки 503 «Service Temporarily Unavailable» [38].

Для верифікації роботи механізму необхідно відкрити веб-додаток за адресою localhost:3000, сторінка завантажиться у звичайному режимі. Однак, якщо спробувати виконати швидке оновлення сторінки декілька разів поспіль (інтервал менше 2 секунд), користувач побачить сторінку з повідомленням про помилку 503. Це є індикатором того, що захисний механізм спрацював коректно, і сервер відхилив надлишкові запити.

Конфігурація віртуальних хостів

У контексті адміністрування веб-сервісів поняття віртуального хостингу є ключовим механізмом, що дозволяє одному фізичному або віртуальному серверу обслуговувати декілька незалежних доменних імен одночасно. Ця технологія базується на здатності веб-сервера аналізувати HTTP-заголовки, зокрема заголовок Host, для маршрутизації вхідного запиту до відповідного кореневого каталогу файлової системи або специфічного обробника. Впровадження віртуальних хостів є необхідною умовою для оптимізації використання апаратних ресурсів, оскільки дозволяє уникнути розгортання окремого екземпляра операційної системи для кожного веб-сайту. У середовищі Ubuntu Server реалізація віртуальних хостів для Apache2 та серверних блоків для Nginx має уніфіковану структуру каталогів, проте відрізняється синтаксисом конфігураційних файлів та алгоритмами обробки запитів, що вимагає детального розгляду кожного підходу.

Організація файлової системи є першим етапом у проектуванні мультидоменного середовища. Згідно зі стандартом FHS (Filesystem Hierarchy Standard), дані веб-сайтів зазвичай розміщуються у директорії /var/www/. Для забезпечення ізоляції контенту для кожного домену створюється окрема директорія, наприклад /var/www/example.com/html.

Критично важливим аспектом є налаштування прав доступу та власників файлів. Процес веб-сервера, який у Debian-подібних системах зазвичай виконується від імені користувача www-data, повинен мати права на читання файлів для їх відправки клієнту, а в деяких випадках – і на запис (для директорій завантаження медіа-файлів або кешу). Налаштування здійснюється за допомогою системних утиліт chown для зміни власника та chmod для встановлення маски прав доступу, що забезпечує базовий рівень безпеки та запобігає несанкціонованому доступу між різними віртуальними хостами.

Конфігурація віртуальних хостів у веб-сервері Apache2 базується на використанні модульної системи конфігураційних файлів. У дистрибутиві Ubuntu прийнята схема, де доступні конфігурації зберігаються в каталозі /etc/apache2/sites-available/, а активовані – у /etc/apache2/sites-enabled/. Кожен віртуальний хост описується в окремому файлі з розширенням .conf у блоці <VirtualHost *:80> (для HTTP) або <VirtualHost *:443> (для HTTPS). Ключовою директивою, яка визначає приналежність запиту, є ServerName, де вказується основне доменне ім'я, та ServerAlias, що дозволяє вказати альтернативні імена (наприклад, з префіксом www). Директива DocumentRoot вказує шлях до каталогу з файлами сайту. Важливим елементом адміністрування є розділення журналів подій: директиви ErrorLog та CustomLog дозволяють записувати помилки та статистику доступу в окремі файли для кожного домену, що значно спрощує процес відлагодження та аудиту безпеки [36].

Активація віртуального хоста в Apache2 здійснюється шляхом створення символічного посилання з каталогу sites-available в sites-enabled. У середовищі Ubuntu для цього розроблено спеціалізовану утиліту a2ensite (Apache2 Enable Site), яка автоматизує цей процес. Після виконання команди sudo a2ensite example.com.conf, система створює необхідне посилання. Однак зміни набувають чинності лише після перезавантаження конфігурації веб-сервера. Перед цим обов'язково виконується синтаксична перевірка конфігураційних файлів за допомогою команди apache2ctl configtest або apache2 -t. Якщо перевірка повертає статус Syntax OK, ініціюється м'яке перезавантаження служби командою systemctl reload apache2, що дозволяє застосувати нові налаштування без розриву існуючих активних з'єднань користувачів.

У веб-сервері Nginx концепція віртуальних хостів реалізується через механізм, що називається «Server Blocks» (серверні блоки). Аналогічно до Apache, в Ubuntu використовується структура каталогів /etc/nginx/sites-available/ та /etc/nginx/sites-enabled/. Конфігурація описується директивою server { ... }, всередині якої параметр listen визначає порт прослуховування, а директива server_name вказує доменні імена, на які реагуватиме даний блок [36].

Особливістю Nginx є алгоритм вибору сервера. При надходженні запиту система перевіряє відповідність заголовка Host значенням server_name. Якщо точного співпадіння не знайдено, запит обробляється сервером, позначеним атрибутом default_server у директиві listen, або першим завантаженим конфігураційним файлом. Це вимагає від адміністратора уважності при проектуванні конфігурацій для уникнення ситуацій, коли запити потрапляють на неправильний обробник [38].

Для коректної роботи PHP-додатків у віртуальних хостах Nginx (оскільки Nginx не має вбудованого модуля обробки PHP, як Apache) налаштовується взаємодія з менеджером процесів PHP-FPM через протокол FastCGI. У блоці location ~ \.php$ прописуються параметри передачі запитів до сокету PHP-FPM (наприклад, fastcgi_pass unix:/var/run/php/php8.1-fpm.sock). Також важливо налаштувати порядок обробки індексних файлів через директиву index, надаючи пріоритет index.php перед index.html. Активація конфігурації в Nginx зазвичай виконується шляхом ручного створення символічного посилання командою ln -s, хоча логіка роботи залишається ідентичною до Apache. Перевірка синтаксису здійснюється командою nginx -t, яка аналізує структуру блоків, наявність крапок з комою та коректність шляхів, після чого виконується перезапуск служби.

Окремим аспектом конфігурації віртуальних хостів є забезпечення безпеки передачі даних за допомогою протоколу HTTPS. Це передбачає додавання у конфігураційний файл окремого блоку VirtualHost (Apache) або server (Nginx), що прослуховує порт 443. У цьому блоці обов'язково вказуються шляхи до SSL-сертифіката та приватного ключа за допомогою

директив SSLCertificateFile / SSLCertificateKeyFile (для Apache) або ssl_certificate / ssl_certificate_key (для Nginx) [38].

Сучасною практикою є налаштування автоматичного перенаправлення всього трафіку з порту 80 на порт 443 для примусового використання шифрованого з'єднання. Це реалізується шляхом створення спрощеного віртуального хоста на порту 80, який повертає HTTP-код 301 з новою адресою, що гарантує цілісність та конфіденційність обміну даними між клієнтом та сервером.

Безпека веб-сервісів

Забезпечення інформаційної безпеки веб-сервісів у середовищі Linux є комплексним, багаторівневим процесом, що вимагає системного підходу до захисту конфіденційності, цілісності та доступності даних. В умовах постійного зростання кількості кіберзагроз, адміністратори серверів повинні реалізовувати стратегію «глибинного захисту, яка передбачає створення ешелонованих рубежів захисту: від рівня операційної системи та мережевого стеку до рівня додатків і баз даних.

Основним етапом забезпечення безпеки є проведення регулярного аудиту стану операційної системи для виявлення потенційних слабких місць у конфігурації та невідповідностей стандартам безпеки. Для автоматизації цього процесу в професійному середовищі широко застосовується спеціалізований інструментарій, зокрема утиліта Lynis. Це програмне забезпечення призначене для глибокого аналізу системи, перевірки цілісності файлів, аналізу встановленого програмного забезпечення та конфігураційних файлів на предмет відповідності кращим практикам безпеки. Процес імплементації даного інструменту в середовищі Ubuntu Server розпочинається з його інсталяції через стандартний менеджер пакетів. Для цього виконується команда з правами суперкористувача sudo apt install lynis –y, яка завантажує необхідні бінарні файли та залежності, інтегруючи утиліту в систему.

Після успішної інсталяції програмного забезпечення ініціюється процедура базового аудиту системи. Запуск перевірки здійснюється за допомогою команди sudo lynis audit system. Під час виконання цієї операції Lynis проводить сканування сотень параметрів системи. Перевіряються права доступу до критично важливих файлів, налаштування завантажувача GRUB, параметри ядра Linux, статус мережевих інтерфейсів, налаштування SSH-сервера та веб-серверів (Apache/Nginx). Алгоритм роботи програми базується на порівнянні поточного стану системи з базою знань про відомі вразливості та стандарти безпеки (наприклад, ISO 27001 або PCI-DSS). Результатом роботи утиліти є деталізований звіт, що виводиться безпосередньо в термінал та записується у лог-файли [38].

Важливим етапом роботи з результатами аудиту є аналіз отриманих даних. Адміністратор повинен ретельно вивчити секції звіту, зосереджуючись на повідомленнях категорій «Warning» (Попередження) та «Suggestion» (Пропозиція). Попередження зазвичай вказують на серйозні недоліки, такі як відсутність пароля на завантажувачі, некоректні права на файли конфігурації або наявність застарілого програмного забезпечення, що потребує негайного виправлення. Пропозиції носять рекомендаційний характер і спрямовані на подальше посилення захисту, наприклад, активацію додаткових модулів ядра або налаштування банерів безпеки. Систематичне виправлення виявлених недоліків дозволяє значно підвищити так званий «індекс безпеки» сервера, мінімізуючи поверхню атаки.

Наступним рівнем захисту веб-сервісів є протидія атакам методом повного перебору (brute-force атаки), які спрямовані на підбір паролів до служб віддаленого доступу (SSH) або панелей адміністрування веб-додатків. Для автоматизації захисту від подібних загроз в інфраструктурі Linux використовується система запобігання вторгненням Fail2Ban. Цей сервіс функціонує шляхом постійного моніторингу лог-файлів (системних журналів) обраних служб. При виявленні підозрілої активності, яка відповідає заданим шаблонам (наприклад, багаторазові невдалі спроби входу з однієї IP-адреси), Fail2Ban динамічно змінює правила міжмережевого екрану (iptables або nftables), тимчасово або постійно блокуючи доступ для джерела атаки. Інсталяція сервісу виконується командою sudo apt install fail2ban –y [38].

Після інсталяції пакета необхідно забезпечити автоматичний запуск служби при завантаженні системи та її негайну активацію. Це досягається послідовним виконанням команд управління системними службами: sudo systemctl enable fail2ban для додавання в

автозавантаження та sudo systemctl start fail2ban для запуску служби. Архітектура конфігурації Fail2Ban передбачає використання файлів .conf для стандартних налаштувань, які можуть бути перезаписані при оновленні пакету, та файлів .local для користувацьких налаштувань. Тому, згідно з кращими практиками адміністрування, редагування основного конфігураційного файлу jail.conf не рекомендується. Замість цього створюється або редагується файл локальних правил jail.local.

Налаштування правил захисту веб-сервера здійснюється шляхом редагування файлу конфігурації через текстовий редактор, наприклад, командою sudo nano /etc/fail2ban/jail.local. У цьому файлі описуються так звані «ізолятори» для конкретних сервісів. Для веб-серверів Apache або Nginx налаштовуються секції, що відстежують логи помилок та логи доступу. Визначаються параметри bantime (час блокування порушника), findtime (інтервал часу, за який підраховуються невдалі спроби) та maxretry (максимальна допустима кількість спроб перед блокуванням). Коректно налаштований файл jail.local дозволяє ефективно відсікати автоматизовані бот-мережі, що сканують сервер на наявність вразливостей, не впливаючи при цьому на легітимний трафік. Після внесення змін служба перезапускається для застосування нових правил.

Окрім внутрішнього аудиту та захисту, важливим аспектом безпеки є зовнішнє тестування периметра мережі, яке імітує дії потенційного зловмисника. Для виявлення відкритих портів, доступних сервісів та версій програмного забезпечення використовується мережевий сканер Nmap. Цей інструмент дозволяє адміністратору побачити свій сервер «очима хакера» і виявити непотрібні відкриті служби, які можуть слугувати точкою входу для атаки. Перед використанням утиліту необхідно інсталювати в систему командою sudo apt install nmap –y. Регулярне сканування дозволяє контролювати дотримання принципу мінімізації привілеїв, згідно з яким доступними ззовні мають бути лише критично необхідні порти (зазвичай 80, 443 та порт SSH).

Процедура тестування вразливостей ініціюється запуском Nmap з певними параметрами сканування. Команда sudo nmap -sV -p 1-1000 <IP-серверу> виконує сканування першої тисячі портів вказаного хоста. Параметр -sV є ключовим у даному контексті, оскільки він активує режим визначення версій служб (Service Version Detection). Це дозволяє отримати інформацію не лише про те, що порт відкритий, а й про те, яка саме служба його прослуховує (наприклад, Apache 2.4.41 або Nginx 1.18.0). Знання точної версії ПЗ є критичним для аналізу безпеки, оскільки дозволяє перевірити наявність відомих CVE (Common Vulnerabilities and Exposures) для даної версії. Якщо сканування виявляє порти, які не повинні бути публічними (наприклад, порти баз даних 3306 або службові порти 8080), адміністратор повинен негайно закрити їх за допомогою брандмауера UFW.

Важливою складовою безпеки веб-сервісів є також налаштування міжмережевого екрану. В Ubuntu Server стандартним є використання UFW – інтерфейсу для спрощеного управління правилами iptables. Політика безпеки за замовчуванням має бути налаштована на заборону всіх вхідних з’єднань (default deny incoming) та дозвіл вихідних (default allow outgoing). Дозвіл на вхідні з’єднання надається виключно для необхідних протоколів: SSH (порт 22 або змінений), HTTP (80) та HTTPS (443). Це створює надійний бар’єр, що запобігає несанкціонованому доступу до внутрішніх системних служб, які можуть бути активовані за замовчуванням, але не призначені для публічного використання [38].

Захист даних під час передачі забезпечується використанням криптографічних протоколів SSL/TLS. Сучасний стандарт безпеки вимагає обов’язкового використання HTTPS для всіх веб-ресурсів. У середовищі Linux для цього широко використовується інструмент Certbot, який дозволяє автоматизувати процес отримання та оновлення безкоштовних сертифікатів від центру сертифікації Let’s Encrypt. Налаштування веб-сервера повинно включати відключення застарілих версій протоколів (SSLv3, TLS 1.0, TLS 1.1) та слабких алгоритмів шифрування. Крім того, рекомендується впровадження механізму HSTS (HTTP Strict Transport Security), який примушує браузери використовувати виключно захищене з’єднання, запобігаючи атакам типу «downgrade» та перехопленню сесій.

Додатковий рівень безпеки реалізується через налаштування спеціальних HTTP-заголовків у конфігурації веб-сервера (Nginx або Apache). Заголовки, такі як X-Frame-Options

(запобігає атакам Clickjacking), X-Content-Type-Options (забороняє браузеру змінювати MIME-типи файлів) та Content-Security-Policy (CSP) (обмежує джерела завантаження контенту, захищаючи від XSS-атак), є обов'язковими для сучасних веб-додатків. Впровадження CSP вимагає ретельного аналізу роботи додатку, оскільки некоректна політика може порушити функціональність сайту, блокуючи легітимні скрипти або стилі [38].

Варто зазначити, що тільки комплексне поєднання методів захисту дозволяє гарантувати стабільну та безпечну роботу веб-сервісів у мережевому середовищі.

Журнали і моніторинг

Ефективне адміністрування серверної інфраструктури неможливе без реалізації комплексної стратегії спостережуваності, яка складається з двох фундаментальних компонентів: журналювання подій (логування) та моніторингу стану системи.

Журнали забезпечують історичний контекст, фіксуючи дискретні події, що відбулися в минулому, тоді як моніторинг надає інформацію про поточний стан ресурсів та метрик продуктивності в реальному часі. У середовищі Ubuntu Server ці процеси реалізуються через взаємодію системних служб, ядра операційної системи та спеціалізованих утиліт аналізу.

Центральним елементом підсистеми журналювання в сучасних дистрибутивах Linux, включаючи Ubuntu, є служба systemd-journald. Вона перехоплює повідомлення від ядра, служб ініціалізації, стандартного виводу процесів (stdout/stderr) та системної служби журналювання syslog. На відміну від традиційних текстових файлів, journald зберігає дані у бінарному структурованому форматі, що забезпечує індексацію та швидкий пошук. Доступ до цих даних здійснюється через утиліту journalctl. Адміністратор має можливість фільтрувати події за часом (наприклад, journalctl --since «1 hour ago»), за конкретним сервісом (параметр -u, наприклад, journalctl -u nginx) або за пріоритетом повідомлення. Для перегляду подій у режимі реального часу використовується прапор -f (follow), що дозволяє відслідковувати нові записи в момент їх появи, що є аналогом команди tail -f для текстових файлів [36].

Паралельно з journald у системі функціонує служба rsyslog, яка забезпечує сумісність із класичним стандартом syslog та відповідає за запис текстових лог-файлів у директорію /var/log. Основним файлом, де агрегується більшість системних повідомлень, є /var/log/syslog (або /var/log/messages у деяких дистрибутивах). Події, пов'язані з механізмами автентифікації та авторизації, включаючи спроби входу через SSH або виконання команд через sudo, записуються у файл /var/log/auth.log. Аналіз цього файлу є обов'язковою процедурою при розслідуванні інцидентів безпеки. Ядро операційної системи веде власний кільцевий буфер повідомлень, вміст якого можна переглянути командою dmesg або у файлі /var/log/kern.log. Це джерело інформації є першочерговим при діагностиці проблем з апаратним забезпеченням, драйверами пристроїв або мережевим стеком на низькому рівні.

Окремої уваги заслуговує управління дисковим простором, що займають журнали. Оскільки сервери функціонують у режимі постіної роботи, обсяг текстових логів може зростати безконтрольно, що загрожує переповненням файлової системи. Для вирішення цієї проблеми в Linux використовується механізм ротації журналів, що реалізується утилітою logrotate. Конфігурація ротації визначається у файлі /etc/logrotate.conf та директорії /etc/logrotate.d/. Процес ротації передбачає періодичне перейменування поточного файлу журналу (наприклад, у syslog.1), його архівування (компресію у syslog.2.gz) та створення нового порожнього файлу для запису актуальних подій. Ця процедура виконується автоматично за розкладом планувальника завдань cron, що забезпечує стабільність використання дискового простору без втручання адміністратора.

У контексті веб-сервісної інфраструктури, налаштованої на базі Apache2 та Nginx, журналювання набуває специфічної структури. Кожен веб-сервер веде два основних типи журналів: журнал доступу та журнал помилок. Для Apache стандартним розташуванням є /var/log/apache2/access.log та /var/log/apache2/error.log. Журнал доступу фіксує кожен HTTP-запит, що надходить до сервера, зберігаючи IP-адресу клієнта, час запиту, метод (GET, POST тощо), запитуваний ресурс, код відповіді статусу (наприклад, 200 або 404) та інформацію про клієнтський агент. Аналіз журналу доступу дозволяє будувати статистику відвідуваності, виявляти патерни поведінки користувачів та ідентифікувати джерела аномального трафіку.

Журнал помилок, своєю чергою, містить детальну інформацію про внутрішні збої сервера, помилки в скриптах веб-додатку або проблеми з конфігурацією, що робить його основним інструментом відлагодження для розробників та системних адміністраторів [36].

Для моніторингу та отримання миттєвого зрізу стану системи в терміналі використовується утиліта top або її більш функціональний аналог htop. Ці інструменти відображають список активних процесів, відсортованих за споживанням процесорного часу або оперативної пам'яті, а також загальні показники навантаження системи (Load Average). Показник Load Average, що складається з трьох значень (за 1, 5 та 15 хвилин), відображає середню кількість процесів, що очікують на виконання або введення-виведення. Якщо це значення перевищує кількість доступних процесорних ядер, це свідчить про перевантаження системи та необхідність оптимізації або масштабування ресурсів.

Для детального аналізу підсистеми пам'яті використовується команда free -h, яка показує загальний обсяг фізичної пам'яті, обсяг використаної та вільної пам'яті, а також розмір буферів та кеш-пам'яті. Важливо розуміти, що Linux агресивно використовує вільну оперативну пам'ять для дискового кешування з метою пришвидшення роботи системи, тому низьке значення стовпця «free» при високому значенні «available» є нормальною поведінкою. Для діагностики проблем з дисковою підсистемою застосовується утиліта iostat (з пакету sysstat), яка дозволяє виявити вузькі місця у швидкості читання/запису на накопичувачі, а команда df -h надає інформацію про доступний простір на змонтованих файлових системах. Вичерпання дискового простору є типовою причиною аварійної зупинки сервісів.

Моніторинг мережевої активності у Linux є критично важливим для забезпечення доступності веб-сервісів та деяких інших ролей сервера (DNS, DHCP та ін.). Для візуального моніторингу в середовищі з графічним інтерфейсом відкриваються «Додатки», там здійнсюється пошук групи додатків «Система», і там обирається «System Monitor» (Системний монітор). У вікні «Системного монітора» здійнюється перехід на вкладку «Resources» (Ресурси). У розділі «Мережа» відображаються: швидкість прийому/передачі даних на сервері (RX/TX) та загальний обсяг переданих даних. Це надає загальну картину завантаженості мережевого інтерфейсу. Також у вкладці «Processes» (Процеси) можна відсортувати процеси та подивитися, які програми активно використовують мережу.

У професійному адмініструванні серверів, де графічний інтерфейс часто відсутній, моніторинг мережевої активності здійснюється переважно у терміналі. Для аналізу відкритих портів та встановлених з'єднань використовується команда ss. Активні з'єднання переглядаються командою ss -tulpn. Дана команда показує активні сокети (TCP/UDP), стан з'єднання (LISTEN, ESTABLISHED), локальні та віддалені адреси й процеси, що ініціювали з'єднання. Це дозволяє швидко виявити несанкціоновані служби або перевірити, чи коректно веб-сервер прослуховує необхідні порти [36].

Для моніторингу трафіку конкретного мережевого інтерфейсу в реальному часі встановлюється в терміналі інструмент «iftop». Це виконується командою sudo apt install iftop –y. Після цього запускається даний інструмент – sudo iftop -i enp0s3. Замість «enp0s3» використовується актуальна назва інтерфейсу, яку можна дізнатися командою ip addr. Утиліта iftop візуалізує поточні з'єднання між локальним хостом та віддаленими адресами, відображаючи смугу пропускання для кожного з'єднання окремо, що дозволяє ідентифікувати клієнтів, які генерують найбільше трафіку.

У випадках, коли необхідно визначити, який саме локальний процес споживає мережевий трафік, стандартних засобів може бути недостатньо. Для моніторингу трафіку по процесах виконується встановлення «nethogs». Для цього вписується команда sudo apt install nethogs –y і встановлюється даний інструмент. Після цього запускається – sudo nethogs. Ця утиліта групує трафік не за IP-адресами, а за PID (індентифікаторами процесів), показуючи, які процеси створюють навантаження на мережу (швидкість відправки SENT та отримання RECEIVED). Це незамінний інструмент для виявлення завислих скриптів резервного копіювання або скомпрометованих процесів, що генерують паразитний трафік.

Для моніторингу статистики інтерфейсів та помилок в передачі мережевого трафіку на канальному рівні застосовується команда ip -s link. Дана команда виведе детальну статистику по кожному мережевому інтерфейсу, включаючи кількість переданих пакетів, байтів, а також

лічильники помилок, відкинутих пакетів, колізій та переповнень буфера (overrun). Наявність зростаючої кількості помилок у виводі цієї команди часто свідчить про фізичні проблеми з мережевим обладнанням, кабелями або некоректні налаштування дуплексу на комутаторі.

Моніторинг мережевої активності у Linux дозволяє контролювати використання мережевих ресурсів, виявляти аномалії та потенційні загрози, а також оптимізувати продуктивність системи. Використовуючи інструменти як ss, iftop, nethogs та графічні утиліти в GNOME, адміністратор може отримувати детальну інформацію про з'єднання, трафік та активні процеси. Регулярний моніторинг підвищує безпеку та стабільність серверного середовища. Однак, ручний моніторинг через термінал є ефективним лише для оперативної діагностики «тут і зараз». А для побудови довгострокової стратегії обслуговування та моніторингу рекомендується впровадження систем автоматизованого централізованого збору метрик (наприклад, Prometheus у зв'язці з Grafana) та систем агрегації журналів (ELK Stack: Elasticsearch, Logstash, Kibana), які дозволяють автоматизувати виявлення інцидентів та вести постійний моніторинг.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Класифікація комп'ютерних мереж URL: https://mozok.click/736-klasifkacya-kompyuternih-merezh.html (дата звернення: 22.01.2025).
2. Мережні топології. StudFiles. URL: https://studfile.net/preview/7446861/page:9/ (дата звернення: 25.01.2025).
3. Комп'ютерні мережі. Книга 1: навчальний посібник / Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Львів: «Магнолія 2006», 2023. 256 с.
4. Комп'ютерні мережі. Книга 2: навчальний посібник / Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Львів: «Магнолія 2006», 2023. 312 с.
5. Костюченко А.О., Цибко Г.Ю. Адресація в комп'ютерних мережах: навчальнометодичний посібник. URL: http://erpub.chnpu.edu.ua:8080/jspui/bitstream/123456789/7842/1/Adresatsiia_v_kompjuternykh_merezhakh_2021.pdf (дата звернення: 23.02.2025).
6. Introduction To Subnetting - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/computer-networks/introduction-to-subnetting/ (дата звернення: 25.02.2025).
7. Subnetting: Brushing up on the fundamentals. Network World. URL: https://www.networkworld.com/article/969792/subnetting-brushing-up-on-the-fundamentals.html (дата звернення: 26.02.2025).
8. Що таке ARP - Терміни та визначення у сфері кібербезпеки. VPN service for Security and Speed. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/arp?srsltid=AfmBOop5txUSq_BxiwJnJA7Ew74OrAZ-vtWG3E4kszyNR28vv5Fzip6p (дата звернення: 02.03.2026).
9. Networking101Lite Сесія №10 «Динамічна маршрутизація/bgp». URL: http://surl.li/eoivr (дата звернення: 02.03.2025)
10. NetAcademy - Networking101Lite Перша сесія «Модель OSI/Мережі/Базові налаштування обладнання». URL: http://surl.li/eoiux (дата звернення: 02.03.2025)
11. Курс Мережевої академії Cisco CCNA: Introduction to Networks URL: https://www.netacad.com/courses/networking/ccna-introduction-networks (дата звернення: 10.04.2025).Hyper-V virtualization in Windows Server and Windows. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/overview (дата звернення: 02.03.2025).
12. Курс Мережевої академії Cisco CCNA: Switching, Routing, and Wireless Essentials URL: https://www.netacad.com/courses/networking/ccna-switching-routing-wireless-essentials (дата звернення: 02.03.2025).
13. Dauti B. Windows Server 2025 Administration Fundamentals: A beginner's guide to managing and administering Windows Server environments: Fourth Edition. Birmingham : Packt Publishing, 2025. 634 p.
14. System Requirements for Hyper-V on Windows and Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/host-hardware-requirements&pivots=windows (дата звернення: 05.03.2025).
15. Morimoto, R., Noel, M., Yardeni, G., Droubi, O., Abbate, A., & Amaris, C. Windows Server 2012 Unleashed. Indianapolis : Pearson Education. URL: https://api.pageplace.de/preview/DT0400.9780133115970_A23602136/preview-9780133115970_A23602136.pdf (дата звернення: 05.03.2025).
16. Огляд Microsoft Windows Server 2025. URL: https://softlist.com.ua/ua/news/oglyad-microsoft-windows-server (дата звернення: 12.03.2025).
17. Discover what's new in Windows Server 2025. URL: https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/windows-server-2025-data-sheet.pdf (дата звернення: 12.03.2025).
18. Active Directory overview – Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/troubleshoot/windows-server/active-directory/active-directory-overview (дата звернення: 15.03.2025).

19. Install Active Directory Domain Services on Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/deploy/install-active-directory-domain-services--level-100- (дата звернення: 15.03.2025).

20. Install Active Directory Domain Services on Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/deploy/install-active-directory-domain-services--level-100- (дата звернення: 20.03.2025).

21. Active Directory Domain Services overview. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview (дата звернення: 22.03.2025).

22. Group Policy overview for Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/group-policy/group-policy-overview (дата звернення: 28.03.2025).

23. Group Policy preferences in Windows. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/group-policy/group-policy-preferences (дата звернення: 05.04.2025).

24. Group Policy Management Console in Windows. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/group-policy/group-policy-management-console (дата звернення: 14.04.2025).

25. The Admin's Guide to Group Policy Best Practices | Netwrix. Data Security that Starts with Identity| Netwrix. URL: https://netwrix.com/en/resources/guides/group-policy-best-practices/ (дата звернення: 18.04.2025).

26. Group Policy Management Guide. Active Directory Pro. URL: https://activedirectorypro.com/group-policy-guide/ (дата звернення: 22.04.2025).

27. Group Policies and Group Policies Preferences (2025). Hybrid Infrastructure and Cloud Architecture. URL: https://hartiga.de/windows-server/group-policies-foundation/ (дата звернення: 25.04.2025).

28. Install and Configure DNS Server on Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/networking/dns/quickstart-install-configure-dns-server&tabs=powershell (дата звернення: 02.05.2025).

29. Manage DNS zones using DNS server in Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/networking/dns/manage-dns-zones&tabs=powershell (дата звернення: 02.05.2025).

30. What is DHCP Server in Windows Server?. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-top (дата звернення: 12.05.2025).

31. Install and configure DHCP Server on Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/networking/technologies/dhcp/quickstart-install-configure-dhcp-server?tabs=powershell (дата звернення: 16.05.2025).

32. Guidance for troubleshooting DHCP - Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/troubleshoot/windows-server/networking/troubleshoot-dhcp-guidance (дата звернення: 24.05.2025).

33. Migrate existing DHCP failover deployment on Windows Server. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows-server/networking/technologies/dhcp/migrate-existing-dhcp-failover?tabs=powershell (дата звернення: 26.05.2025).

34. Ubuntu Server documentation. Ubuntu Server. URL: https://documentation.ubuntu.com/server/ (дата звернення: 28.05.2025).

35. Munna R. Linux DNS Server Configuration: Detailed Guide [2025]. MailServerGuru. URL: https://mailserverguru.com/linux-dns-server/#Master-Update-the-System (дата звернення: 28.05.2025).

36. Imron M. Guide to Creating a Simple Web Server Using Nginx and Apache2. Medium. URL: https://medium.com/@muhammadimron1410/guide-to-creating-a-simple-web-server-using-nginx-and-apache2-ae7d27b421c6 (дата звернення: 03.06.2025).

37. Microsoft Learn. Server Message Block (SMB) protocol overview. – URL: https://learn.microsoft.com/en-us/windows-server/storage/file-server/file-server-smb-overview (дата звернення: 05.06.2025).

38. Ubuntu Documentation. Setting up Samba as a File Server. – URL: https://ubuntu.com/server/docs/samba-file-server (дата звернення: 10.06.2025).

І-74 Інформаційні мережі та адміністрування: конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Інформаційні системи та технології охорони і безпеки» галузі знань 12/F Інформаційні технології спеціальності 126/F6 Інформаційні системи та технології денної та заочної форм навчання / уклад. Н. В. Багнюк, О. Л. Кайдик. Луцьк: ЛНТУ, 2026. 236 с.

Конспект лекцій з дисципліни «Інформаційні мережі та адміністрування» складено відповідно до діючої програми курсу.

Призначено для здобувачів вищої освіти спеціальності 126/F6 Інформаційні системи та технології освітньої програми «Інформаційні системи та технології охорони і безпеки».

Комп’ютерний набір Н. В. Багнюк

Редактор Н. В. Багнюк

Підп. до друку «___» _____ 2026р.

Формат 60х84/16. Папір офс. Гарнітура Таймс.

Ум. друк. арк. ____. Тираж 10 прим. Зам. ____

Відділ іміджу та промоцій

Луцького національного технічного університету

43018, м. Луцьк, вул. Львівська, 75