Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-технічний засіб керування конфігурацією системи домених імен на основі BIND
Назва теми

КвРКІ 180107.18.01.07 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія»
Назва

Виконав: студент IV курсу, група КІ–18–1 _____ А.Р. Карцан
Підпис Ініціали, прізвище

Керівник _____ С.М. Лисенко
Підпис, дата Ініціали, прізвище

Нормоконтролер _____ С.М. Лисенко
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної інженерії та інформаційних систем _____ Т.О. Говорущенко
Підпис Ініціали, прізвище

«1» 06 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Г.О.Говорущенко

" 11 " 01 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Карцану Артуру Руслановичу

Прізвище, ім`я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб керування конфігурацією системи доменних імен на основі BIND

Керівник проекту (роботи) Лисенко С.М., д.т.н., професор

Прізвище, ім`я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 06.01.2022 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2022 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі

Проектування програмно-технічного засобу

Програмно-апаратна реалізація та тестування програмно-технічного засобу

конфігурацією та інфраструктурою локального центру обробки даних

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Конфігураційні налаштування

Налаштування DNS

Налаштування DNS

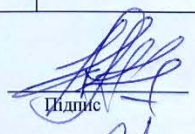
6. Консультанти розділів дипломного проекту (роботи)

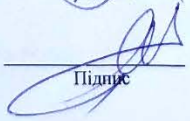
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІІП		
Антиплагіат	Нічепорук А.О., доцент кафедри КІІП		

7. Дата видачі завдання «__06__»_____09_____2021 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2022	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2022	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2022	виконано
4	Робота над розділом 2 – Проектування програмно-технічного засобу	01.04.2022	виконано
5	Робота над розділом 3 - Програмно-апаратна реалізація та тестування програмно-технічного засобу	30.04.2022	виконано
6	Оформлення пояснювальної записки згідно вимог	31.05.2022	виконано
7	Попередній захист ВКР	02.06.2022	виконано
8	Захист ВКР на засіданні ЕК	Червень 2022 року	

Студент  А. Р. Карцан
Підпис Ініціали, прізвище

Керівник проекту (роботи)  С. М. Лисенко
Підпис Ініціали, прізвище

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 180107.18.01.07 ПЗ	Пояснювальна записка	58		
			Графічні матеріали			
2		КвРКІ 180107.18.01.07 Е8	Конфігураційні налаштування	1		
3		КвРКІ 180107.18.01.07 Е8	Налаштування DNS	1		
		КвРКІ 180107.18.01.07 Е8	Налаштування DNS	1		
4						

КвРКІ 180107.18.01.07 ВП

Зм	Арк	№ докум	Підпис	Дата	Відомість проекту	Літера	Аркуш	Аркушів
Розробив		Каршан А.Р.				У	1	1
Перевір.		Лисенко С.М.				ХНУ, КІ-18-1		
Н. контр.		Лисенко С.М.						
Затв.		Говорущенко						

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб керування конфігурацією та інфраструктурою локального центру обробки даних».

Автор роботи: Карцан Артур Русланович.

Керівник роботи: Лисенко Сергій Миколайович.

Пояснювальна записка: 58 с., 12 рис., 5 табл., 4 дод., 28 джерел.

Графічна частина: 15 презентаційних слайдів.

КОМП'ЮТЕРНА МЕРЕЖА, DNS-СЕРВЕР, BIND, IT ІНФРАСТРУКТУРА, ЦЕНТР ОБРОБКИ ДАНИХ.

Метою роботи є розробка програмно-технічного засобу керування конфігурацією системи доменних імен на основі BIND.

У цій роботі розроблений програмно-технічний засіб керування конфігурацією системи доменних імен на основі BIND.

Підпис студента  Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОС - операційна система

ПЗ - програмне забезпечення

КС – комп’ютерна система

КМ – комп’ютерна мережа

ВСТУП

DNS-сервіс є одним із важливих сервісів для нормального функціонування Інтернет-мережі.

Його основне завдання полягає в визначення відповідності між мережними адресами вузлів мережі та їх зручно читабельними назвами.

Існує два варіанти визначення цієї відповідності - пряме та реверсивне визначення. При прямому дозволі DNS-сервер на ім'я визначає і видає мережеву адресу, а при реверсивному – за адресою шукає відповідне ім'я.

Це необхідно враховувати при настроюванні DNS-сервісу, оскільки для здійснення Дані механізми використовуються різні таблиці.

В операційній системі Sun Solaris, як, в іншому, та інших UNIX-системах, як DNS-сервер використовується BIND-сервер версії 8.x та вище.

Хоча, слід зазначити, у Solaris є можливість використання та сервера версії 4.x. Система визначає яку версію DNS-сервер запускати по тому, який конфігураційний файл існує в каталозі / etc.

Якщо використовується файл – named.boot, то запускається стара версія сервісу, а якщо - named.conf - то відповідно нова.

Краще звичайно використовувати BIND 8.x і вище. Якщо у вас залишилися файли конфігурації named.boot і є необхідність перевести DNS-сервер на нову версію, то можна скористатися скриптом /usr/sbin/named-bootconf який конвертує конфігураційний файл BIND 4.x BIND 8.x.

Технологія використання "strong name" обчислює кеш або контрольну суму, що підписується складання і тим самим гарантує її справжність для клієнтського програми (або майже гарантує).

Технологія використання "strong name" визначається правилами використання криптографічних ключів у конкретній організації.

В роботі подано основні аспекти керування конфігурацією та інфраструктурою локального центру обробки даних з використанням сервісу BIND.

В основі роботи було розглянуто процес конфігурування BIND-сервера, який складається з двох етапів: налаштування конфігураційного файлу та створення та заповнення таблиць доменних зон.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Програмного забезпечення для взаємодії з системою доменних імен

BIND - це набір програмного забезпечення для взаємодії з системою доменних імен (DNS). Його найпомітніший компонент виконує основні ролі DNS-сервера, діючи як авторитетний сервер для зон DNS і як рекурсивний резольвер в мережі. Станом на 2021 рік це найбільш широко використовуване програмне забезпечення сервера доменних імен [2-4] і є де-факто стандартом для Unix-подібних операційних систем. [5-6] Набір також містить різні інструменти адміністрування, такі як nsupdate і dig, а також бібліотеку інтерфейсу розпізнавання DNS.

Програмне забезпечення було спочатку розроблено в Каліфорнійському університеті Берклі (UCB) на початку 1980-х років. Назва походить як акронім від Berkeley Internet Name Domain , [7] що відображає використання програми в UCB. Остання версія – BIND 9, вперше випущена в 2000 році і досі активно підтримується Консорціумом Інтернет-систем (ISC), а нові випуски виходять кілька разів на рік.

BIND призначений для повної відповідності стандартам IETF DNS і чернеткам стандартів . Важливі функції BIND 9 включають: TSIG , nsupdate , IPv6, RNDC (контроль віддаленого демона імен), представлення даних, підтримку багатопроцесорів, обмеження швидкості відповіді (RRL), DNSSEC та широку переносимість. RNDC дозволяє віддалено оновлювати конфігурацію, використовуючи загальний секрет для забезпечення шифрування для локальних і віддалених терміналів під час кожного сеансу.

У той час як попередні версії BIND не передбачали жодного механізму для зберігання та отримання даних про зону в будь-якому іншому, окрім плоских текстових файлів, у 2007 BIND 9.4 [8] DLZ надав опцію часу компіляції для зберігання зон у різноманітних форматах баз даних, включаючи LDAP , Berkeley DB , PostgreSQL , MySQL і ODBC.

BIND 10 планував зробити сховище даних модульним, щоб можна було підключати різноманітні бази даних. [9] У 2016 році ISC додала підтримку інтерфейсу dyndb, наданого RedHat, з BIND версії 9.11.0. [10]

Проблеми безпеки, виявлені в BIND 9, виправляються та публічно розкриваються відповідно до загальних принципів програмного забезпечення з відкритим кодом. Повний список дефектів безпеки, які були виявлені та розкриті в BIND9, підтримується Консорціумом Internet Systems, поточними авторами програмного забезпечення [11].

Обидва випуски BIND 4 і BIND 8 мали серйозні вразливості безпеки. Використання цих стародавніх версій або будь-якої непідтримуваної версії, яка не підтримується, настійно не рекомендується [12]. BIND 9 був повністю переписаним , частково для пом'якшення цих проблем безпеки. Сторінка завантажень на веб-сайті ISC чітко показує, які версії наразі обслуговуються, а які закінчилися.

1.2 Основні компоненти BIND

Основна мета DNS - це відображення доменних імен в IP-адреси і навпаки - IP в DNS. У статті я розгляну роботу DNS сервера BIND (Berkeley Internet Name Domain, раніше: Berkeley Internet Name Daemon), як самого (не побоюся цього слова) поширеного. BIND входить до складу будь-якого дистрибутива UNIX. Основу BIND складає демон named , який для своєї роботи використовує порт UDP/53 та для деяких запитів TCP/53.

Історично, до появи доменної системи імен роль інструменту дозволу символьних імен в IP виконував файл /etc/hosts, що й у час грає далеко ще не останню роль цьому справі. Але зі зростанням кількості хостів у глобальній мережі, відстежувати та обслуговувати базу імен на всіх хостах стало неможливо. В результаті придумали DNS, що є ієрархічною, розподіленою системою доменних зон.

Доменна структура DNS є деревоподібною ієрархією, що складається з вузлів, зон, доменів, піддоменів та ін елементів, про які нижче йтиметься мова.

«Вершиною» доменної структури є коренева зона. Налаштування кореневої зони розташовані на безлічі серверів/дзеркал, розміщених по всьому світу і містять інформацію про всі сервери кореневої зони, а також відповідають за домени першого рівня (ru, net, org та ін). Інформація про сервери кореневої зони розташована на цьому сайті кореневих серверів . Налаштування кореневої зони завжди доступні тут . Сервери кореневої зони обробляють та відповідають на запити, надаючи інформацію лише про домени першого рівня (тобто відповідають на будь-які запити, як на нерекурсивні)! Отже, багато разів повторилося слово зона. Час цей термін пояснити.

Зона - це будь-яка частина дерева системи доменних імен, що розміщується як єдине ціле на деякому DNS-сервері . Зону, для більшого розуміння, можна назвати «зоною відповідальності» . Метою виділення частини дерева в окрему зону є передача відповідальності (Делегування) за цю галузь іншій особі чи організації. На ілюстрації, приклади зон виділені синім градієнтом (зона name , зона k-max.name . З усіма підлеглими ресурсами, www.openoffice.org з усім підлеглими піддоменами та ресурсами). На ілюстрації виділені в повному обсязі зони, лише деякі для загального розуміння і уявлення. У кожній зоні є принаймні один авторитетний сервер DNS , який зберігає ВСЮ інформацію про зону, за яку він відповідає.

Домен - це іменована гілка або піддерево в дереві імен DNS, тобто це певний вузол, що включає всі підлеглі вузли. Наступна цитата з книги Linux Network Administrators Guide добре прояснює картину щодо різниці між зоною та доменом:

Таким чином, простір імен роздроблений на зони (zones), кожна з яких управляється своїм доменом. Зверніть увагу на різницю між зоною (zone) та доменом (domain): домен groucho.edu зачіпає всі машини в університеті Groucho Marx, в той час як зона groucho.edu включає тільки хости, які працюють безпосередньо у комп'ютерному центрі, наприклад у відділі математики . Хост у відділі фізики належить іншій зоні, а саме physics.groucho.edu.

Кожен вузол в ієрархії DNS відокремлений від свого батька крапкою. Якщо провести аналогію з файловою системою Linux, система доменних імен має схожу

структуру, за тим винятком, що роздільник у файловій системі - слеш , а в DNS - точка . А також DNS адреса читається праворуч наліво (від кореневого домену до імені хоста) на відміну від шляху у файловій системі Linux. Доменне ім'я починається з точки (кореневого домену) і проходить через домени першого, другого і якщо потрібно третього і т.д. рівнів та завершується ім'ям хоста. Т.ч. доменне ім'я повністю відображає структуру ієрархії DNS . Часто (я б сказав - завжди в повсякденному житті), остання точка (позначення кореневого домену) в доменному імені опускається (тобто в браузері ми не вводимо k-max.nam e. , а k-max.nam e). Отже, розібравши структуру доменного імені, ми непомітно наблизилися до поняття FQDN .

FQDN (англ. Fully Qualifed Domain Name , повністю певне ім'я домену) - це ім'я домену, однозначно визначає доменне ім'я і включає імена всіх батьківських доменів ієрархії DNS, в тому числі і кореневого . Своєрідний аналог абсолютного шляху у файловій системі.

Відмінність між FQDN і стандартним доменним (неFQDN) ім'ям утворюється при іменуванні доменів другого, третього (і т. д.) рівня. Для отримання FQDN потрібно обов'язково вказати в доменні імені домени вищого рівня (наприклад, mail є доменним ім'ям, однак ім'я FQDN виглядає як mail.k-max.name.). Максимальний розмір FQDN - 255 байт, з обмеженням 63 байта на кожне ім'я домену.

Піддомени , коротко кажучи, це підлеглі домени . За великим рахунком всі домени в інтернеті є підлеглими за винятком кореневого. Наприклад, домен k-max є піддоменом домену name, а name, у свою чергу - піддоменом кореневого домену.

Отже, на схемі вище ми розглянули кореневий домен , наступним в ієрархії йдуть домени першого/верхнього рівня , вони ж TLD , вони ж Top-Level Domain . До даних доменів відносяться національні домени (ru. , ua. та ін) та загальні домени (com. , net. , та ін). Існують також спеціалізовані домени , які не опубліковані в системі DNS, але використовуються програмами (домен .onion використовується анонімною мережею Tor для перехоплення та подальшої маршрутизації звернень до прихованих сервісів цієї мережі). Ще є т.зв.

зарезервовані доменні імена, визначені в RFC 2606 (Reserved Top Level DNS Names — Зарезервовані імена доменів верхнього рівня) визначає назви доменів, які слід використовувати як приклади (наприклад, документації), а також для тестування. До таких імен відносяться наприклад mysite, example.org і example.net, а також test, invalid та ін. Нижче по ієрархії, очевидно, йдуть домени третього рівня і т.д. Закінчується доменна ієрархія - іменами хостів , які задаються відповідними ресурсними записами або хостовими записами .

Ресурсний запис - це те, що заради чого в кінцевому рахунку і існує DNS. Ресурсний запис – це одиниця зберігання та передачі інформації в DNS. Кожна така запис несе інформацію відповідності якогось імені та службової інформації в DNS, наприклад відповідність імені домену — IP адреси.

Запис ресурсу складається з наступних полів.

Ім'я (NAME) — доменне ім'я, до якого прив'язана або якому належить цей ресурсний запис, або IP адреса. За відсутності даного поля запис ресурсу успадковується від попереднього запису.

Time To Live (TTL) - дослівно "час життя" запису, час зберігання запису в кеші DNS (після вказаного часу запис видаляється), дане поле може не вказуватися в індивідуальних записах ресурсів, але тоді воно має бути вказано на початку файлу зони і буде успадковуватися усіма записами.

Клас (CLASS) — визначає тип мережі, (у 99,99% випадках використовується IN (що означає — Internet). Дане поле було створено з припущення, що DNS може працювати і в інших типах мереж, крім TCP/IP)

тип (TYPE) — тип записи синтаксис и назначение записи

дані (DATA) — різна інформація, формат та синтаксис якої визначається типом.

При цьому можна використовувати такі символи:

; - Вводить коментар

- Також вводить коментарі (тільки у версії BIND 4.9)

@ — Ім'я поточного домену

( ) — Дозволяють даним займати кілька рядків

* - Метасимвол (тільки в полі ім'я)

З усім набором ресурсних записів можна ознайомитись у wikipedia . Найчастіше застосовувані ресурсні записи такими (далі, ми обов'язково розглянемо їх практично):

A - (address record/запис адреси) відображають ім'я хоста (доменне ім'я) на адресу IPv4. Для кожного мережного інтерфейсу машини має бути зроблено один A-запис .

AAAA (IPv6 address record) аналогична записи A, но для IPv6.

CNAME (canonical name record/канонічний запис імені (псевдонім)) - Відображає аліас на реальне ім'я (для перенаправлення на інше ім'я.

MX (mail exchange) - Вказує хости для доставки пошти, адресованої домену. У цьому полі NAME вказує домен призначення, поля TTL , CLASS — стандартне значення, поле TYPE набуває значення MX , а полі DATA вказує пріоритет і через пробіл - доменне ім'я хоста, відповідального прийому пошти . Наприклад, наступний запис показує, що для домену k-max.name надсилайте пошту спочатку на mx.k-max.name, потім на mx2.k-max.name, якщо з mx.k-max.name виникли якісь проблеми .

NS (name server/сервер імен) вказує на DNS-сервер, який обслуговує цей домен. Вірніше буде сказати — вказують сервери, на які делеговано цей домен. Якщо записи NS відносяться до серверів імен для поточної зони, доменна система їх практично не використовує. Вони просто пояснюють, як організована зона і які машини відіграють ключову роль забезпеченні сервісу імен.

PTR (pointer) — відображає IP-адресу в доменне ім'я (про цей тип запису поговоримо нижче в розділі зворотного перетворення імен).

SOA (Start of Authority/початковий запис зони) - описує основні / початкові налаштування зони, можна сказати, визначає зону відповідальності даного сервера . Для кожної зони має існувати лише один запис SOA і вона має бути першою. Поле Name містить ім'я домену/зони , поля TTL, CLASS — стандартне значення, поле TYPE набуває значення SOA , а поле DATA складається з кількох значень, розділених пробілами: ім'я головного DNS (Primary Name Server) , адреса адміністратора зони , далі в дужках — серійний номер файлу зони (Serial number) . При кожному внесенні змін до файлу зони це

значення необхідно збільшувати, це вказує вторинним серверам, що зона змінена, і що їм необхідно оновити зону. Далі - значення таймерів (Refresh - вказує, як часто вторинні сервери повинні опитувати первинний, щоб дізнатися, чи не збільшився серійний номер зони, Retry - час очікування після невдалої спроби опитування, Expire - максимальний час, протягом якого вторинний сервер може використовувати інформацію про отриманій зоні, Minimum TTL – мінімальний час, протягом якого дані залишаються в кеші вторинного сервера).

1.3 Висновки

В розділі представлено основні аспекти функціонування програмного забезпечення для взаємодії з системою доменних імен.

Також в розділі описано основні компоненти BIND як сервісу, що дозволяє реалізувати програмно-апаратна реалізація та тестування програмно-технічного засобу керування конфігурацією системи домених імен на основі BIND.

2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

2.1 Встановлення DNS-сервер

Щоб встановити DNS-сервер, необхідно запустити YaST і вибрати Програмне забезпечення › Програмне забезпечення Менеджмент.

Далі необхідно вибрати режим Перегляду та вибрати параметри DHCP і DNS-сервер .

Підтвердьте встановлення залежних пакетів, щоб завершити процес встановлення.

Як альтернатива, скористайтеся такою командою в командному рядку:

```
sudozypper в шаблоні -t dhcp_dns_server
```

2.2 Конфігурація з YaST

Можна використати модуль YaST DNS, щоб налаштувати DNS-сервер для локальної мережі.

Під час першого запуску модуля запускається майстер із підказкою прийняти кілька рішень щодо адміністрування сервера.

Завершення це початкове налаштування створює базову конфігурацію сервера.

Можна використати експерта режим для вирішення більш складних завдань конфігурації, наприклад налаштування ACL, журналювання, ключі TSIG та інші параметри.

2.3 Майстер конфігурації

Майстер складається з трьох кроків або діалогів. У відповідних місцях у діалогах ви можете увійти в режим експертної конфігурації.

При першому запуску модуля Forwarder налаштувань.

Місцева Політика дозволу DNS дозволяє встановити наступні параметри:

1.Об’єднання пересилань вимкнено.

2. Автоматичне злиття.

3. Об’єднання пересилань увімкнено.

Спеціальна конфігурація – якщо Користувацька конфігурація, то пункт Спеціальна політика можна вказати за замовчуванням (з автоматичним злиттям), для спеціальної політики встановлено значення auto, але тут можна встановити імена інтерфейсу або вибрати із двох спеціальних назв політики STATICі STATIC_FALLBACK.

У Local DNS Resolution Forwarder необхідно вказати, яка послуга для використання: використання системних серверів імен, сервер імен (bind) або локальний dnsmasq сервер .

Додаткову інформацію про всі ці налаштування див man 8 netconfig. (рисунок 2.1).

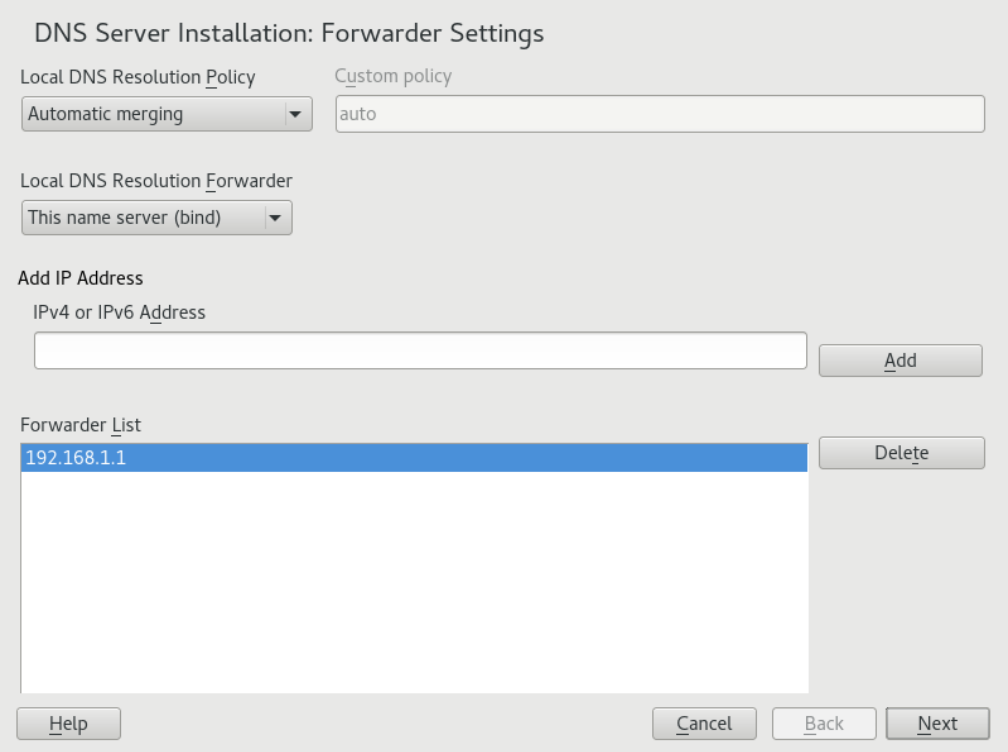


Рисунок 2.1 – Встановлення DNS-сервера

Пересилачі — це DNS-сервери, на які ваш DNS-сервер надсилає йому запити не може сам відповісти.

Для цього необхідно ввести їх IP-адресу та натисніть Додати .

Діалогове зон DNS складається з кількох частин і є відповідає за управління файлами зони.

Для нової зони введіть для неї назву в Ім'я.

Щоб додати зворотну зону, назва має закінчуватися на .in-addr.arpa. Нарешті, виберіть Тип (головний, підпорядкований або прямий).

Необхідно натиснути Редагувати щоб налаштувати інші параметри існуючої зони.

Щоб видалити зону, необхідно клацнути Видалити (рисунок 2.2).

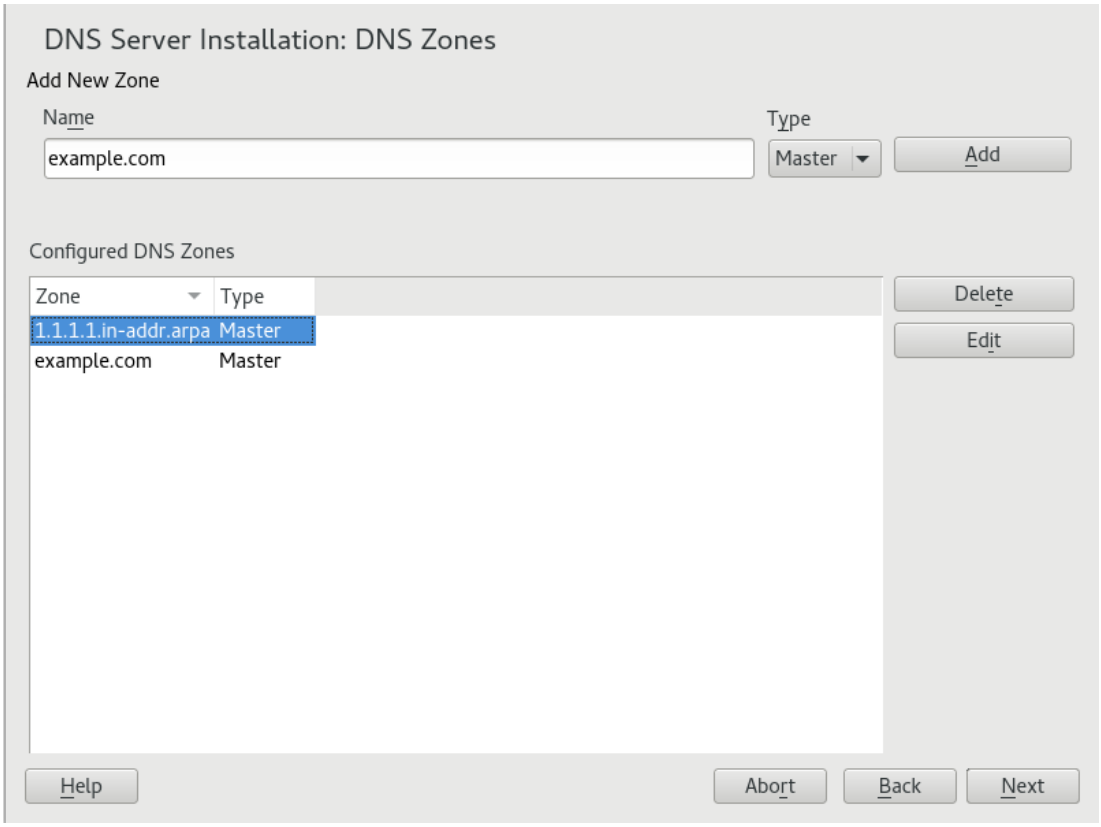


Рисунок 2.2 – Встановлення сервера DNS

У останньому діалоговому вікні можна відкрити порт DNS у брандмауері натиснувши Відкрити порт у брандмауері.

Тоді необхідно запустити DNS-сервер під час завантаження (Увімкнено або Вимкнено).

Також можна активувати підтримку LDAP.

Після запуску модуля YaST відкривається вікно, яке відображає кілька параметри конфігурації.

Завершення цього призведе до конфігурації DNS-сервера з основними функціями:

У «Запуск» необхідно визначити, чи має бути DNS-сервер запускається під час завантаження системи або вручну.

Щоб запустити DNS-сервер необхідно натиснути Запустити DNS-сервер зараз.

Щоб зупинити DNS-сервер, натисніть Зупинити DNS-сервер зараз.

Щоб зберегти поточні налаштування, виберіть

Зберегти налаштування та Перезавантажити DNS-сервер.

Тепер можна відкрити порт DNS у брандмауері за допомогою Open Port in Брандмауер і змініть налаштування брандмауера за допомогою брандмауера.

Необхідно вибрати LDAP Support Active, файли зони будуть керуватися базою даних LDAP.

Будь-які зміни даних зони, записаних до LDAP базу даних забираються DNS-сервером під час його перезапуску або запиту щоб перезавантажити його конфігурацію.

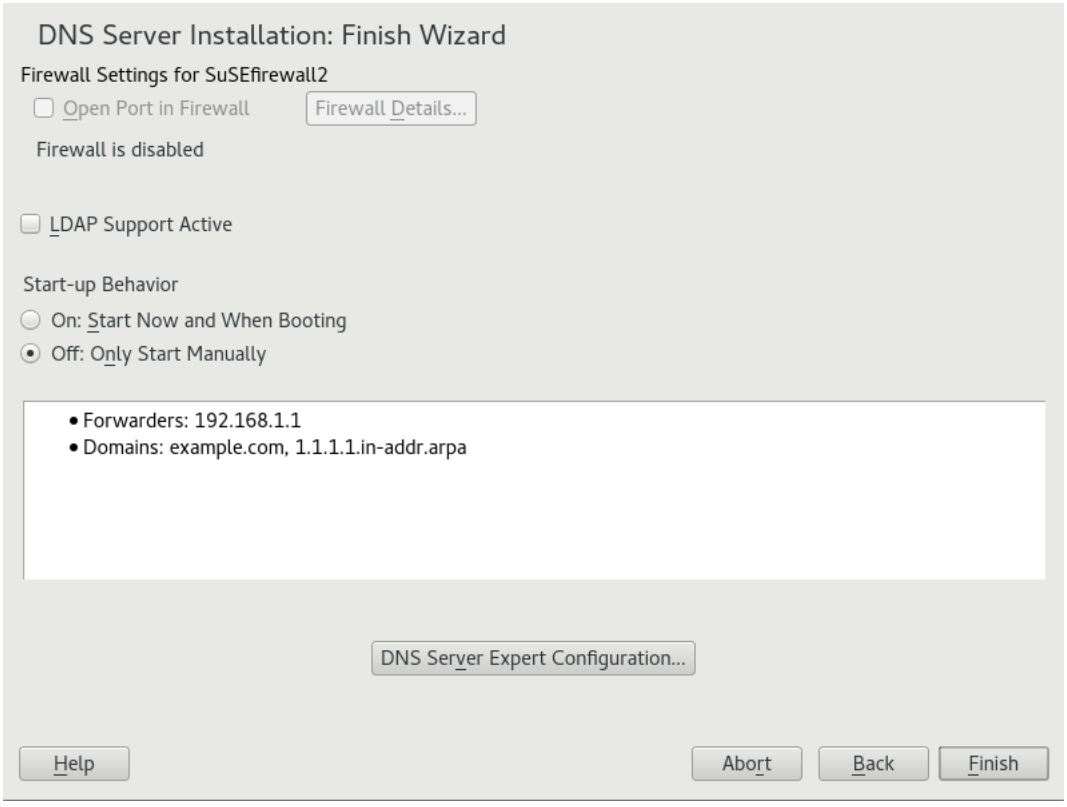


Рисунок 2.3 – Встановлення DNS-сервера

2.4 Налаштування повідомлень про помилки

Якщо локальний DNS-сервер не може відповісти на запит, він намагається переслати запит до експедитора , якщо так налаштовано.

Цей експедитор може бути доданий вручну до списку використовуваних експедиторів.

Якщо пересилач не є статичним, як у комутованих з'єднаннях, netconfig обробляє конфігурацію.

Для більш інформацію про netconfig можна скористатися командою:

man 8 netconfig.

У цьому розділі файлу можна встановити основні параметри сервера.

Від параметрів необхідно вибрати потрібний пункт і вказати його значення у відповідному текстовому полі.

Включіть новий запис, вибравши Додати.

Щоб налаштувати, що і як реєструвати DNS-сервер, виберіть пункт Налатувати.

У розділі Тип журналу необхідно вказати, куди сервер DNS повинен записати дані журналу.

Необхідно вказати загальносистемний журнал виберіть Системний журнал або вкажіть інший файл за вибравши Файл.

В останньому випадку додатково необхідно вказати назву, максимальний розмір файлу в мегабайтах і кількість журналів версії файлів для збереження.

Додаткові параметри доступні в розділі Додатковий журнал.

Увімкнення журналу всіх причин кожен запит, який потрібно зареєструвати, у цьому випадку файл відкритого журналу може стати надзвичайно великим.

З цієї причини необхідно увімкнути цю опцію для інших цілей, ніж для налагодження.

Для реєстрації даних трафіку під час оновлення зони між DHCP і DNS-сервером, увімкніть Оновлення зони журналу.

Для реєстрації трафіку даних під час зони передати з головного на підпорядкований, увімкніть зону журналу Передача.

Можна використати це діалогове вікно, щоб визначити ACL (списки контролю доступу), щоб забезпечити доступ обмеження.

Після надання окремої назви під Ім'я (рисунок 2.4), вкажіть IP-адресу (з мережевою маскою або без неї) під значенням таким чином:

{ 192.168.1/24; }

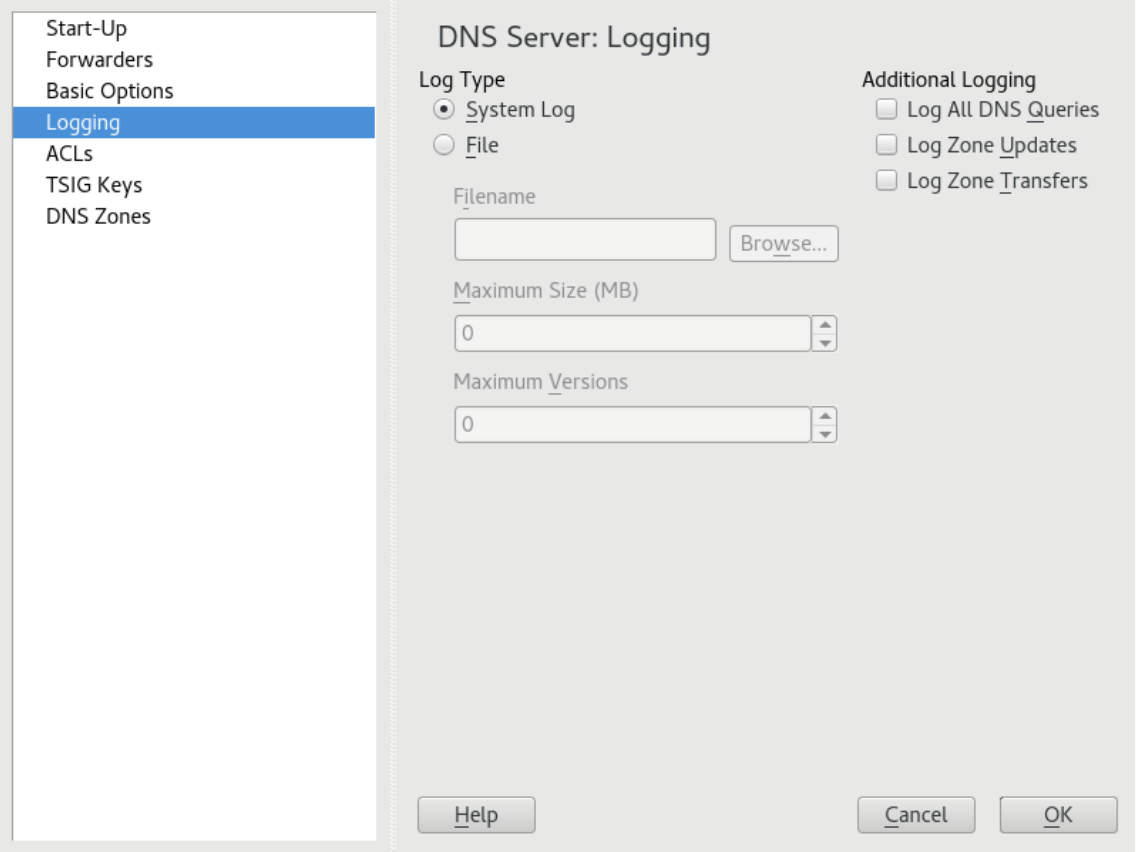


Рисунок 2.4 – DNS-сервер: реєстрація

Синтаксис файлу конфігурації вимагає, щоб адреса закінчувалася на а крапку з комою і поміщають у фігурні дужки.

Основною метою TSIG (підписів транзакцій) є захист зв'язок між серверами DHCP і DNS.

Щоб згенерувати ключ TSIG, введіть відмітне ім’я в поле з міткою Ідентифікатор ключа та вкажіть файл, де має бути збережено згенерований ключ (Ім’я файлу).

Підтвердити свій вибір можна за допомогою кнопки Згенерувати .

Щоб використовувати раніше створений ключ, залиште ID ключа поле пустим і виберіть файл, у якому він зберігається Ім'я файлу.

Після цього необхідно підтвердити за допомогою Додати.

Щоб додати підпорядковану зону, виберіть Зони DNS, необхідно вибрати зону введіть Slave і написати назву нової зони, а також та натиснути Додати.

У Редакторі зон» вікні DNS IP , необхідно вказати головний пункт, з якого ведений має витягнути його дані.

Щоб обмежити доступ до сервера, необхідно вказати один із списків керування доступом у списку список.

Щоб додати головну зону, виберіть Зони DNS, необхідно вказати зону введіть Master , напишіть назву нової зони та клацніть Додати.

При додаванні головної зони також є зворотна зона необхідний. Наприклад, при додаванні зони mysiteщо вказує на хости в підмережі 192.168.1.0/24, також слід додати зворотну зону для охоплений діапазон IP-адрес. За замовчуванням 1.168.192.in-addr.arpa.

Щоб відредагувати головну зону, необхідно вказати Зони DNS , виберіть головну зону з таблиці та натисніть Редагувати.

Діалог складається з кількох сторінок: Основи (відкрита перший), NS Records , MX Records , SOA і Records.

Основне діалогове вікно дає змогу визначити параметри для динамічного DNS та параметри доступу для перенесення зон клієнти та підпорядковані сервери імен.

Щоб дозволити динамічне оновлення зон, необхідно вказати Дозволити динамічні оновлення та відповідний ключ TSIG.

Ключ повинен бути визначений до оновлення починається дія.

Щоб увімкнути передачу зон, необхідно вказати відповідні списки керування доступом. ACL мають бути вже визначені.

У Основи » необхідно вказати, чи вмикати зону трансфери. Використовуйте перелічені списки керування доступом, щоб визначити, хто може завантажувати зони (рисунок 2.5).

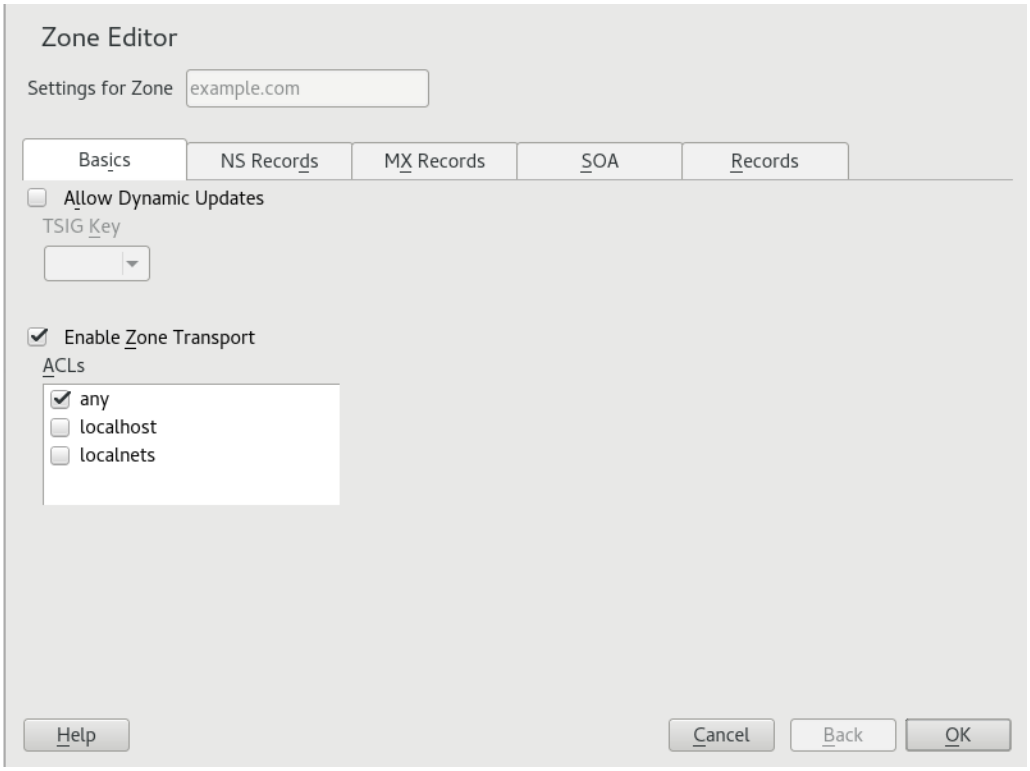


Рисунок 2.5 – Редактор зони

2.5 Редактор зони (NS Records)

Діалогове NS Records дозволяє визначити альтернативні сервери імен для вказаних зон.

Переконайтеся, що власний сервер імен включений до списку.

Щоб додати запис, необхідно ввести його ім’я в розділі Сервер імен для додавання, а потім необхідно підтвердити за допомогою Додати (рисунок 2.6).

Щоб додати поштовий сервер для поточної зони до існуючого списку, необхідно вказати відповідну адресу та значення пріоритету.

Після цього необхідно вибрати Додати (рисунок 2.7).

2.6 Редактор зон (SOA)

Ця сторінка дозволяє створювати записи SOA (початок повноважень). Зміна SOA записи не підтримуються для динамічних зон, керованих через LDAP (рисунок 2.8).

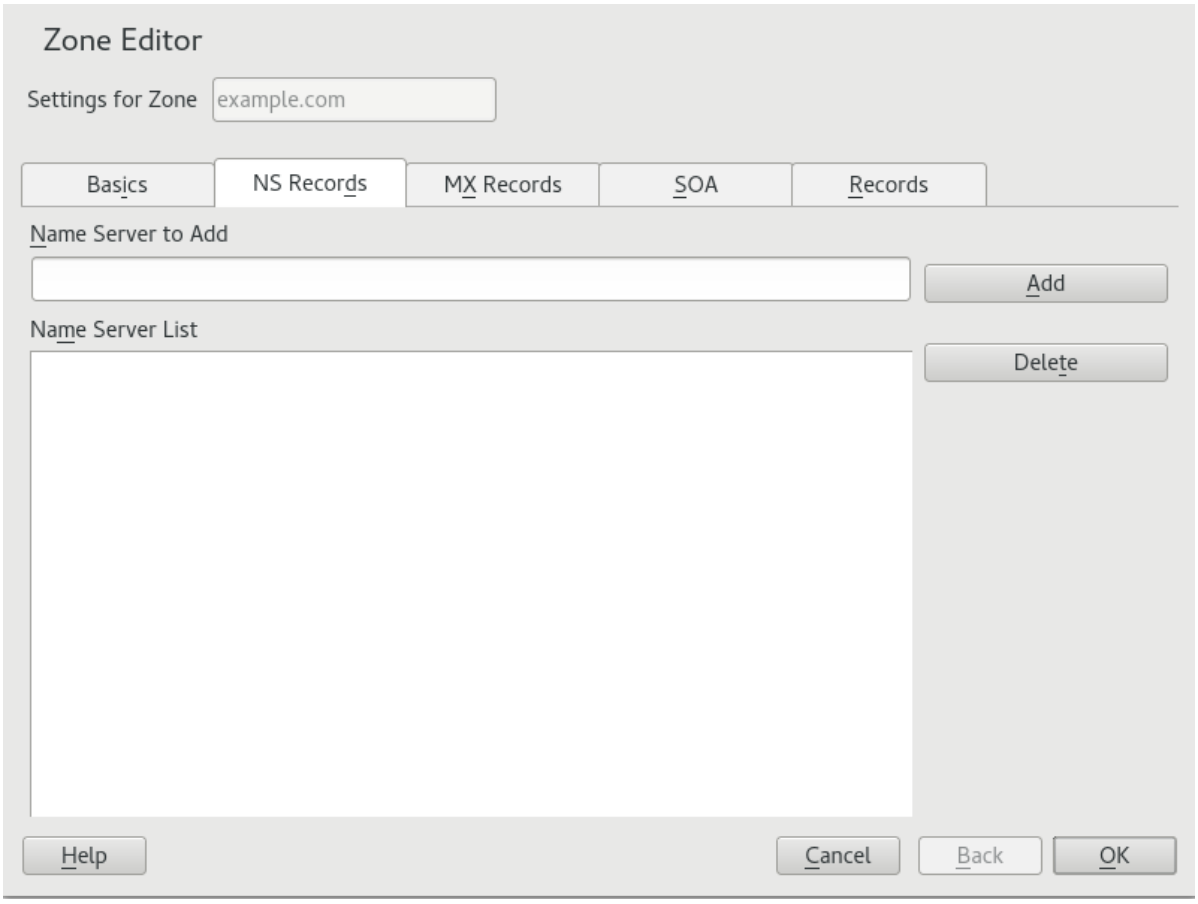


Рисунок 2.6 – Редактор зони (записи NS)

Це діалогове вікно керує розділенням імен. У ключі запису , введіть ім'я хоста, а потім виберіть його тип.

Тип A представляє основний запис. Значенням для цього має бути IP-адреса (IPv4). Можна використати AAAA для адрес IPv6. CNAME – це заданий псевдонім.

Необхідно використати типи NS і MX для детального або часткового записи, які розширюють інформацію, надану в NS Записи та MX Вкладки Ці троє типи розв’язуються до існуючого Азапис. PTR для реверсних зон. Це протилежність A запис, наприклад:

hostname. mysite. B A 192.168.0.1

1.0.168.192.in-addr.arpa IN PTR hostname. mysite.

19.3.2.9.1 Додавання зворотних зон

Щоб додати зворотну зону, виконайте цю процедуру: запустити YaST › DNS Сервер › Зони DNS .

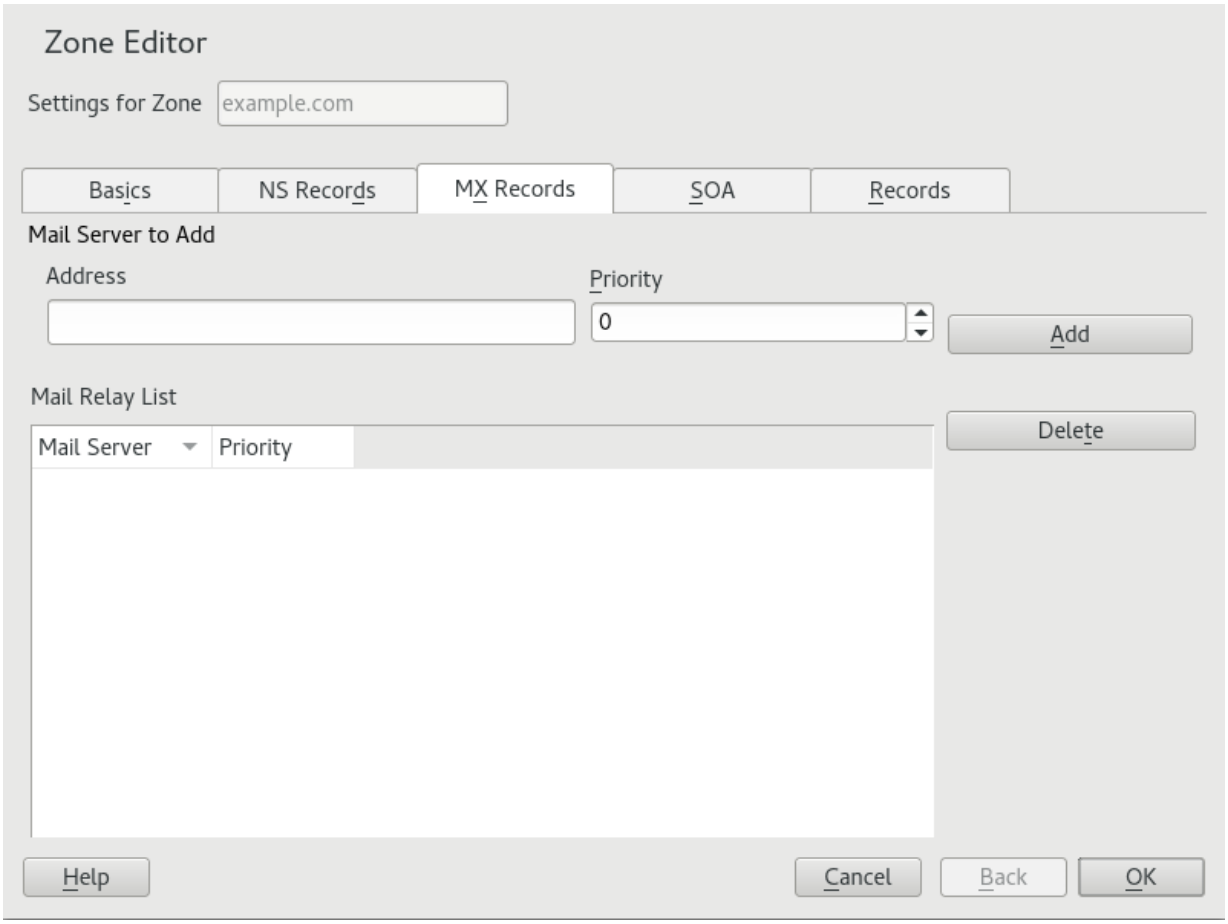


Рисунок 2.7 – DNS-сервер: редактор зони (записи MX)

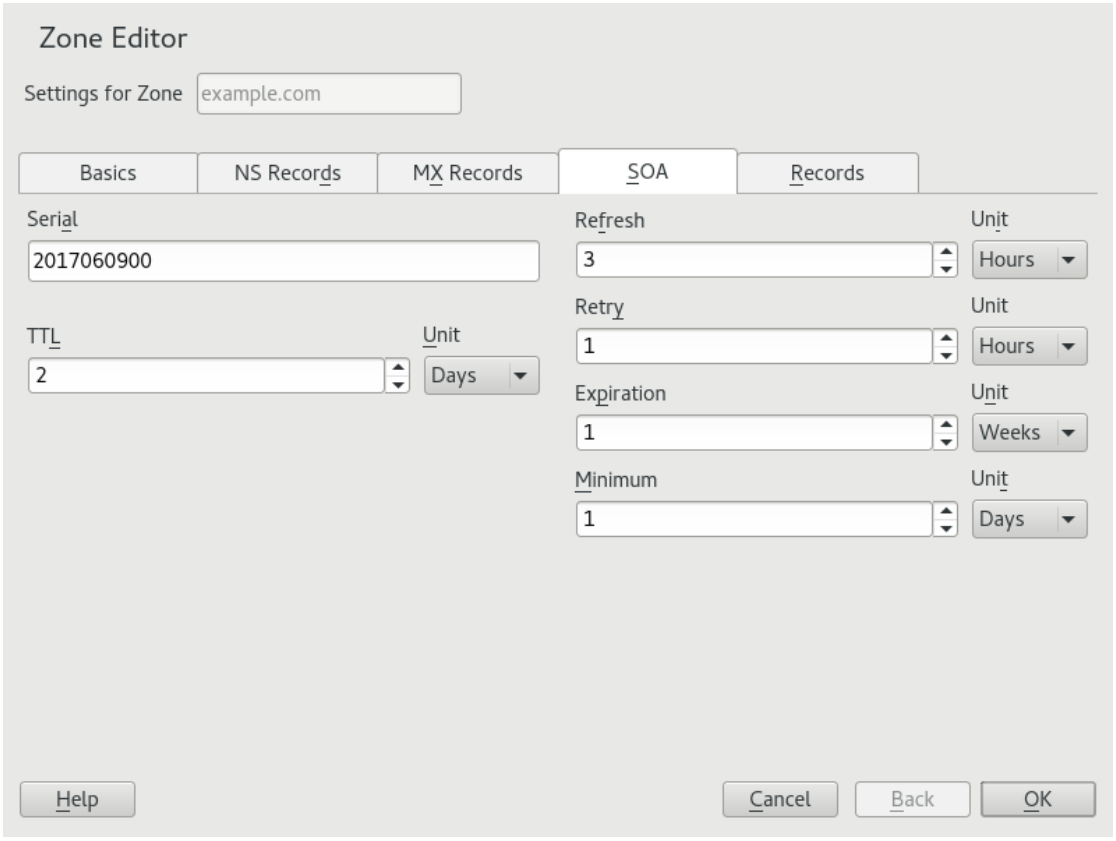


Рисунок 2.8 – Редактор зони (SOA)

Якщо не додано головну пряму зону, необхідно додати її і обов'язково відредагувати.

Потом треба заповнити відповідну функцію, записати ключ і значення, а потім додати запис за допомогою кнопки Додати та підтвердити за допомогою кнопки ОК.

Якщо YaST скаржиться на неіснуючий запис для сервера імен додайте його на NS Records вкладку (рисунок 2.9).

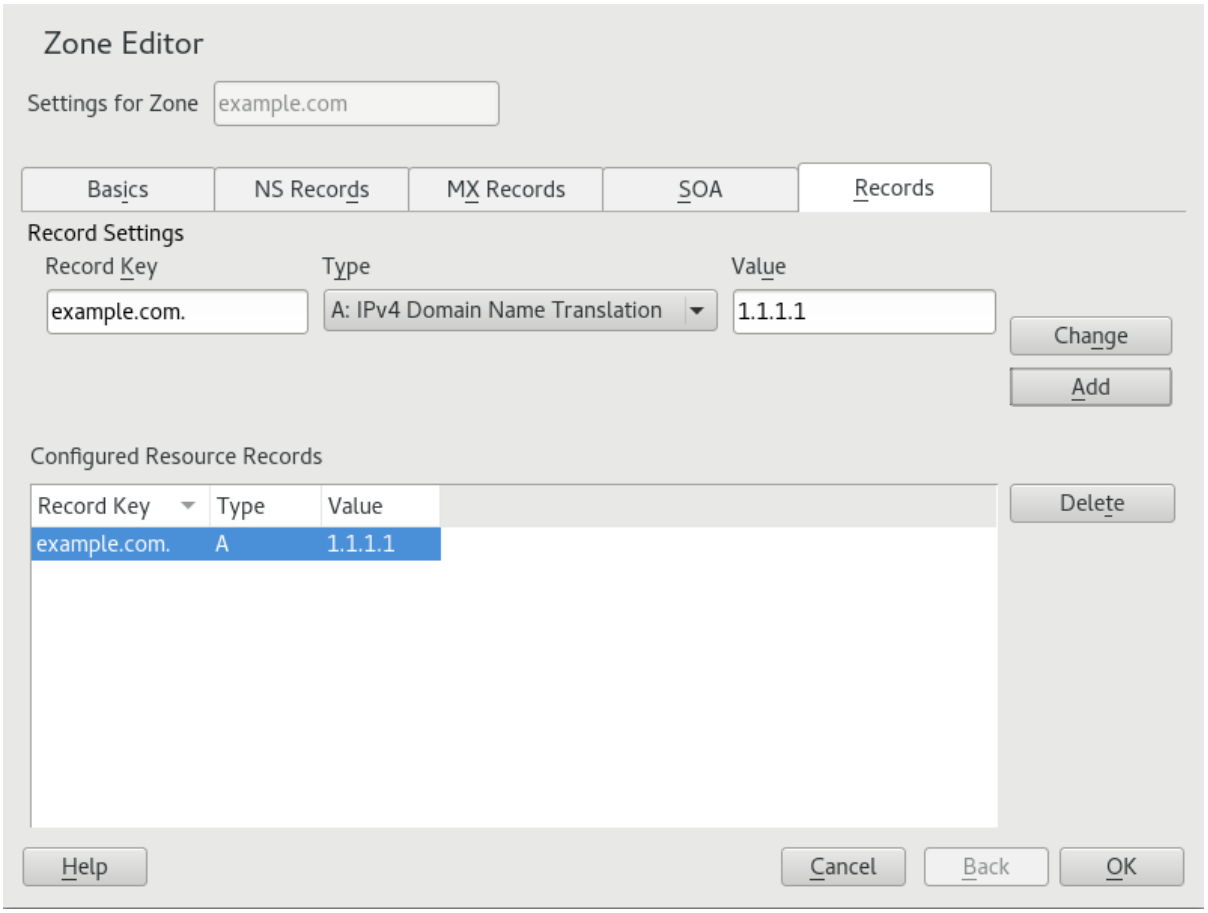


Рисунок 2.9 – Додавання запису

Повернувшись у вікно DNS Zones, можна виконати додавання зворотної головної зони.

Потім необхідно відредагувати зворотну зону і в записах можна побачити PTR: реверс перекладу - тип запису.

Далі треба додати відповідний Запишіть ключ і значення, а потім натиснути Додати та підтвердити, натиснувши кнопку ОК (рисунки 2.10, 2.11).

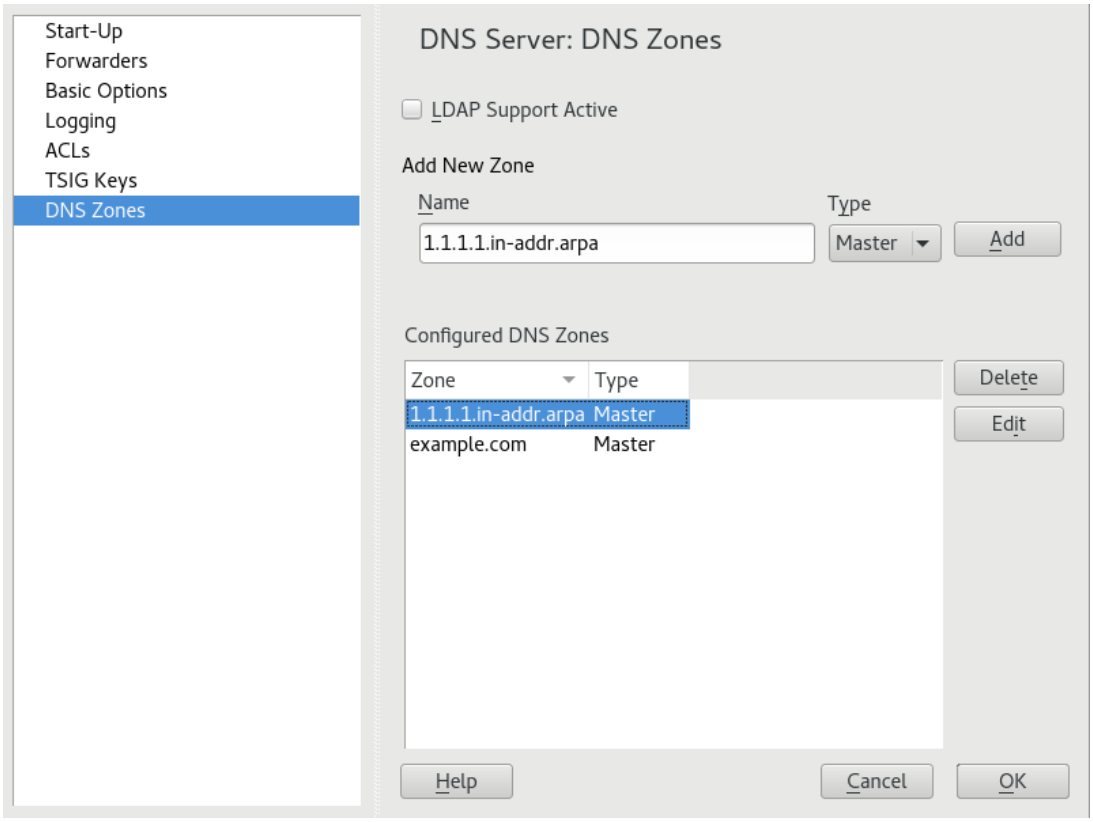


Рисунок 2.10 – Зворотна зона

За потреби можна додайте запис сервера імен.

Після додавання прямої зони необхідно повернутися до головного меню та вибрати зворотну зону для редагування.

Там на вкладці Основи необхідно активувати прапорець Автоматично створювати записи і вибрати пряму зону. Таким чином, все змінюється автоматично.

У системі openSUSE® Leap сервер імен BIND (Berkeley Internet Name Domain) попередньо налаштовано, тому його можна запустити відразу після встановлення.

Зазвичай, якщо вже є підключення до Інтернету і здійснено вхід за адресою 127.0.0.1 як сервер імен адреса для localhostв /etc/resolv.conf, вже є робочий дозвіл імен без необхідності знати DNS постачальника.

Кнопка ЗВ'ЯЗАТИ виконує розділення імен через кореневий сервер імен, що значно повільніше процес.

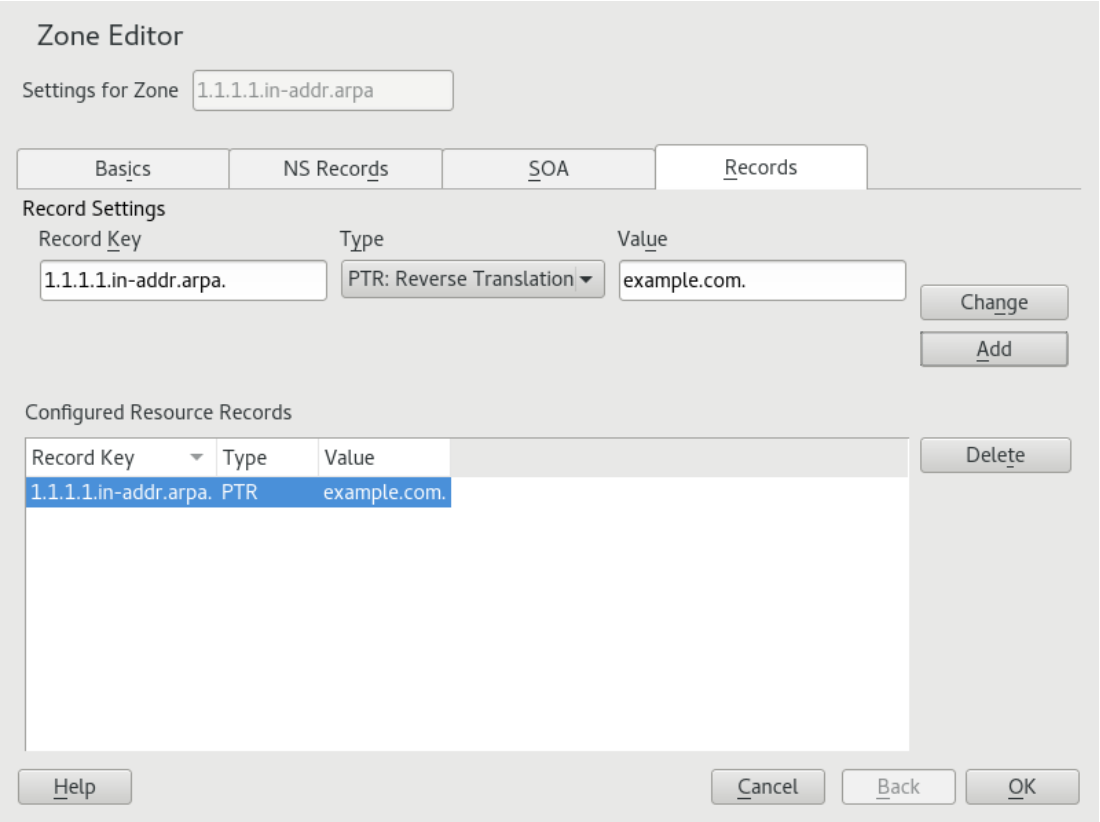


Рисунок 2.11 – Зворотній запис

Зазвичай в DNS провайдера слід вводити його IP адресу у файлі конфігурації /etc/named.conf forwarders щоб забезпечити ефективне та безпечне ім’я.

Якщо це працює, сервер імен працює як чистий лише сервер імен з кешуванням.

Тільки тоді, коли для сервер налаштовано його власні зони, він стає відповідним сервер DNS.

Залежно від типу підключення до Інтернету або підключення до мережі, Інформацію про сервер імен можна автоматично адаптувати до поточних умов. Для цього необхідно встановити змінну NETCONFIG_DNS_POLICY в /etc/sysconfig/network/configфайл до auto.

Не потрібно створювати офіційний домен, доки його не призначить відповідальна установа.

Навіть якщо у є власний домен і він є керований постачальником, краще не використовувати його, тому що BIND не пересилатиме запити для цього домену. Веб-серверне буде доступним для цього домену.

Щоб запустити сервер імен, необхідно ввести команду systemctl start namedяк root.

Далі необхідно перевірии сервер чи успішно запущено командою systemctl status named названий.

Також необхідно перевірити назву сервера в локальній системі.

Якщо це не так, то в /etc/resolv.confмабуть міститься неправильний запис сервера імен або файл не існує.

Для тесту потрібно ввести host 127.0.0.1, який повинен працювати завжди. Якщо отримується повідомлення про помилку, треба використати команду systemctl status named щоб перевірити, чи працює сервер.

Якщо сервер імен не запускається або веде себе несподівано, потрібно перевірити лог journalctl -e.

Щоб використовувати сервер імен (або той, який уже працює в мережі), необхідно ввести відповідну IP-адресу або адреси в розділі option.

Розглянемо налаштування параметрів пересилання в файлі named.conf

```
parameters {
 directory "/ var / lib / named";
 freight forwarders {10.11.12.13; 10.11.12.14; };
 listening {127.0.0.1; 192.168.1.116; };
 allow request {127/8; 192,168 / 16};
 report no;
  };
```

За записом options слідують записи для зони localhost, і 0.0.127.in-addr.arpa. Відповідні файли не потрібно змінювати, вони повинні працювати як є. Також необхідно переконатися, що кожен запис закритий символом «;» і фігурними дужками.

Після зміни конфігурації файл /etc/named.conf або файли зони, необхідно перезапустити за допомогою команди systemctl reload named.

Можна досягнути того ж результату щляхом зупинки та перезапуску сервера імен за допомогою команди systemctl restart named.

Зупинка сервера у будь-який момент можлива за допомогою команди systemctl stop named.

Усі налаштування самого сервера імен BIND зберігаються в файлі /etc/named.conf.

Однак дані зони для домену (складаються з імен хостів, IP-адрес тощо) зберігаються в окремих файлах у /var/lib/named.

Розглянемо налаштування базового звіту.

```
parameters {
 directory "/ var / lib / named";
 freight forwarders {10.0.0.1; };
 report no;
  };
  localhost zone in {
   master type;
   localhost.zone file;
  };
  zone "0.0.127.in-addr.arpa" in {
 master type;
 file "127.0.0.zone";
  };
  zone "." in {
 type hint;
 root.hint file;
  };
```

Для успішного налаштування необхідно розглянути параметри конфігурації:

"FILENAME" directory; вказує каталог, у якому BIND може знайти файли, що містять дані зони. Зазвичай це /var/lib/named.

forwarders { IP-ADDRESS; }; вказує сервери імен (переважно постачальника), до яких DNS запити слід пересилати, якщо вони не можуть бути вирішені безпосередньо. Замініть IP-ADDRESS на IP-адресу, наприклад 192.168.2.120.

forwarders first; спричиняє пересилання запитів DNS через кореневі сервери імен.

Замість forward first, forward onlyможна написати, щоб усі запити пересилалися, а не надсилався на кореневі сервери імен. Це має сенс для конфігурацій брандмауера.

listen-on port 53 { 127.0.0.1; IP-ADDRESS; }; вказує BIND, на якому мережевому інтерфейсі та порту приймати клієнта запити. port 53 не потрібно вказувати явно, тому що 53 є портом за замовчуванням. Введіть 127.0.0.1, щоб дозволити запити від локального хоста.

Якщо повністю опустити цей запис, усі інтерфейси використовуються за замовчуванням.

listen-on-v6 port 53 {any; }; повідомляє BIND, на якому порту він повинен слухати запити клієнта IPv6.

Єдина альтернатива anynone. Що стосується IPv6, сервер приймає лише підстановку адреси.

query-source address * port 53; цей запис необхідний, якщо брандмауер блокує вихідні запити DNS.

Це вказує BIND надсилати запити зовнішньо з порту 53, а не з порту будь-який з високих портів вище 1024.

query-source-v6 address * port 53; вказує BIND, який порт використовувати для запитів IPv6.

query-source-v6 address * port 53; визначає мережі, з яких клієнти можуть надсилати запити DNS. Замінити NET з адресною інформацією, наприклад 192.168.2.0/24.

allow-transfer ! *;; визначає, які хости можуть здійснювати запит щодо перенесення зон.

Без цього запису можна запитувати перенесення зони з будь-якого місця без обмежень.

tatistics-interval 0; за відсутності цього запису BIND створює кілька рядків статистичні відомості за годину в системному журналі.

Встановіть 0 до повністю придушити цю статистику або встановити інтервал у хвилинах.

cleaning-interval 720; визначає, через які інтервали часу BIND очищає кеш. Це запускає запис у системному журналі щоразу, коли це відбувається. Час специфікація в хвилинах. За замовчуванням – 60 хвилин.

cleaning-interval 720; BIND регулярно шукає в мережевих інтерфейсах нові чи неіснуючі інтерфейси.

Якщо це значення встановлено на 0, це є не виконано, і BIND прослуховує лише інтерфейси, виявлені під час запуску.

В іншому випадку інтервал можна визначити в хвилинах. За замовчуванням шістдесят хвилин.

notify no; забороняє іншим серверам імен отримувати інформацію, коли зміни вносяться в дані зони або при перезапуску сервера імен.

Як відбувається ведення журналу, можна детально налаштувати. Зазвичай параметрів за замовчуванням має бути достатньо:

```
logging {
  category default { null; };
};
```

2.7 Налаштування зон

Розглянемо принцип налаштування зон.

```
logging {
  category default { null; };
};
```

Після zone, вказується ім’я домену (mysite.com) слідом за ним блок відповідних опцій укладені у фігурні дужки. Щоб визначити підпорядковану зону,

необхідно перемкнути туредо slavei та вказати сервер імен, який адмініструє цю зону як master:

```
zone "example.net" in {
  type slave;
  file "slave/example.net.zone";
  masters { 10.0.0.1; };
};
```

Існують такі варіанти зон:

type master; зона обробляється локальним сервером імен. Це припускає, що файл зони був створений у правильному форматі.

type slave; зона передається з іншого сервера імен. Його необхідно використовувати разом з masters.

type hint; зона використовується для встановлення кореневих серверів імен. Це визначення зони можна залишити як є

файл mysite.zoneабо файл “ slave/example.net.zone ”; запис визначає файл, у якому знаходяться дані зони для домену. Цей файл не потрібен для підпорядкованого пристрою, оскільки ці дані витягуються з нього інший сервер імен. Щоб розрізнити головні та підпорядковані файли, використовуйте файл каталог slaveдля підпорядкованих файлів.

masters { SERVER_IP_ADDRESS ; }; запис потрібен лише для підпорядкованих зон. У ньому вказується, з якого імені сервера, файл зони слід передати.

allow update {! *; }; параметр контролює зовнішній доступ до запису, що дозволить клієнтам це зробити запис DNS, що не бажано для безпеки. Без цього запису оновлення зони заборонені. Вище входження досягає того ж, оскільки ! *фактично забороняє будь-яку подібну діяльність.

Існують два типи файлів зон. Імена хостів призначають IP-адреси а інший робить навпаки: він надає ім'я хоста для IP-адреси.

Порада: використання крапки (крапка, точка) у файлах зон

Символ "."має важливе значення у файлах зони. Якщо імена хостів наводяться без кінцевої крапки (.), зона додається. Повні імена хостів, указані з

повним іменем домену, повинні закінчуватися з крапкою (.), щоб уникнути додавання домену до нього знову. Відсутній або неправильно розміщений "." це, мабуть, найчастіша причина помилок конфігурації сервера імен.

```
$TTL 2D
 mysite. IN SOA dns  root. mysite. (
 2003072441  ; serial
 1D ; refresh
 2H ; retry
 1W ; expiry
 2D )  ; minimum
 IN NS dns
 IN MX 10 mail dns
gate   IN A  192.168.4.3
 IN A  10.0.0.1
dns IN A  192.168.1.118
mail   IN A  192.168.5.10
jupiter   IN A  192.168.3.10
venus  IN A  192.168.3.11
saturn IN A  192.168.3.12
mercury   IN A  192.168.3.103
ntp IN CNAME dns
dns6   IN A6  0 2022:cea8:111::
```

$TTL визначає час за замовчуванням для цього має застосовуватися до всіх записів у цьому файлі. У цьому прикладі записи дійсні протягом двох днів (2 D).

Запис SOA:

1. Ім'я домену для адміністрування mysitена першій позиції. На цьому закінчується з ".", бо інакше зона була б додано вдруге. Крім того, @може бути введено тут, у цьому випадку зона буде вилучена з відповідний запис в /etc/named.conf.

2. Після IN SO ім'я сервера імен вважається як головне для цієї зони. Назва розширена з dnsdo dns. mysite, тому що так не закінчується на а ".".
3. Далі вказано адресу електронної пошти особи, яка відповідає за цей сервер імен. Тому що @знак вже має особливе значення, "."замість цього вводиться тут. Для root@ mysitezапис має читатися root. mysite.. The "."повинні бути включені в кінці, щоб запобігти зоні від додавання.

В запис SOA включено:

serial numbere 10-значне число. Це повинно змінюватися щоразу, коли цей файл змінюється.

Необхідно повідомити вторинні сервери імен (підпорядковані сервери) змін.

Для цього 10 цифр номер дати та номер прогону, записаний як РРРРММДДНН, став звичайний формат (РРРР = рік, ММ = місяць і DD = день. NN – а порядковий номер, якщо ви оновлюєте його більше одного разу в певний день).

refresh rate вказує інтервал часу в які вторинні сервери імен перевіряють зону serial number. У цьому випадку один день.

retry rate вказує інтервал часу в до якого вторинний сервер імен у разі помилки намагається зв’язатися первинний сервер знову. Ось, дві години.

expiration time визначає часові рамки після чого вторинний сервер імен відкидає кешовані дані, якщо вони є не відновив зв'язок з основним сервером.

Останній запис у записі SOA вказує negative caching TTL—час, за який результати не вирішені Запити DNS з інших серверів можуть кешуватися.

IN NS вказує відповідальний сервер імен для цього домену.

Dns поширюється на dns. mysite тому, що воно не закінчується на а ".".

Таких рядків може бути кілька – один для основний і один для кожного додаткового сервера імен.

Якщо notify не встановлено no в /etc/named.conf, усі перелічені тут сервери імен повідомляти про зміни, внесені до даних зони.

Запис MX визначає поштовий сервер, який приймає, обробляє та пересилає електронні листи для домену mysite. число перед іменем хоста є значенням переваги.

Якщо є кількох записів MX, береться поштовий сервер з найменшим значенням спочатку. Якщо доставка пошти на цей сервер не вдається, запис із використовується наступне найменше значення.

Імена наведені без букви "."тому що вони не включають свій домен, тому mysite додається до усіх.

Хосту призначаються дві IP-адреси gate, оскільки має дві активні мережеві карти.

Де адреса хоста є традиційною (IPv4), запис є позначені з A. Якщо адреса є адресою IPv6, запис позначається AAAA.

Запис IPv6 має дещо інший синтаксис, ніж IPv4.

Тому що можливість фрагментації, необхідно надати інформацію про пропущені біти перед адресою.

Щоб заповнити адресу IPv6 потрібну кількість « 0 » , додайте дві крапки в правильний місце за адресою.

```
pluto  AAAA 1345:00C1:EA11::A234:5678:9ABC:D1F0
pluto  AAAA 1345:00D2:EA11::A34:5678:9ABC:D1F0
```

Псевдонім ntp можна використовувати для адресації dns (CNAMEзасоби канонічна назва).

Псевдодомен in-addr.arpa використовується для зворотного пошук IP-адрес в іменах хостів.

Він додається до мережевої частини адреси у зворотному позначенні. Так 192.168 вирішується в 168.192.in-addr.arpa:

```
$TTL 2D
168.192.in-addr.arpa. IN SOA dns.myexam.com. myroot. mysite(
  2003072441   ; serial
  1D  ; refresh
  2H  ; retry
  1W  ; expiry
```

2D) ; minimum

IN NS mydns. mysite.

1.5 IN PTR mygate. mysite.

100.3 IN PTR www.my mysite.

253.2 IN PTR cups.my mysite.

$TTL визначає стандартний TTL, який застосовується до всіх записів тут.

Файл конфігурації повинен активувати зворотний пошук для мережі 192.168.

Дано що зона називається 168.192.in-addr.arpa, це не слід додавати до імен хостів. Тому всі імена хостів є введені у повній формі – із їхнім доменом та з а "."в кінці.

Решта записів відповідають описані для попереднього mysite приклад.

Рядок визначає сервер імен, відповідальний за цю зону. Цього разу, однак ім'я вводиться в повній формі з доменом і а "."в кінці.

Є лише остання частина IP-адреси введено на початку рядка, без символу "."в кінці.

Додавання зони до цього (без .in-addr.arpa) призводить до отримання повної IP-адреси у зворотному порядку.

Зазвичай передача зон між різними версіями BIND має бути можливо без проблем.

Термін динамічне оновлення відноситься до операцій, за допомогою яких записи у файлах зон головного сервера додаються, змінюються або видаляються. Цей механізм описаний в RFC 2136.

Динамічне оновлення налаштовано окремо для кожної зони, додавши необов’язковий allow-updateабо update-policy правило.

Зони для динамічного оновлення не слід редагувати вручну.

Необхідно передати записи для оновлення на сервер за допомогою команди nsupdate.

З міркувань безпеки будь-яке таке оновлення має бути виконується за допомогою ключів TSIG.

Для забезпечення захищеності транзакції можуть необхідно здійснити налаштування за допомогою підписів транзакцій (TSIG) на основі спільні секретні ключі (також звані ключами TSIG).

Захищені транзакції необхідні для зв’язку між різними серверами і для динамічного оновлення даних зони.

Зробити контроль доступу залежним on ключів набагато безпечніше, ніж просто покладатися на IP-адреси.

Для цього необхідно згенерувати ключ TSIG за допомогою такої команд:

```
sudo dnssec -keygen -a hmac-md5 -b 128 -n HOST host1 -host2
```

Це створює два файли з іменами, подібними до цих:

```
key "host1-host2" {  |
 algorithm hmac-md5;
 secret "oHpBLyfgtcZso6yuwxnugRTWugudJMA==";
};
```

Щоб використовувати його для транзакцій, другий файл необхідно перенести на віддалений хост, бажано безпечним способом (наприклад, за допомогою scp).

На віддаленого сервера, ключ має бути включено в /etc/named.confфайл для забезпечення безпечного зв’язку між host1i host2:

```
key host1-host2 {
 algorithm hmac-md5;
 secret "ejIkgfuCyyGJwuhiwuuN3xAteKgg==";
};
```

Необхідно переконатися, що дозволи задано належним чином в /etc/named.confє.

За замовчуванням для цього файлу є 0640, з власником типу root група named.

Як альтернатива, перемістити ключі до додаткового файлу зі спеціально обмеженими дозволами, тобто потім включено з /etc/named.conf. Щоб включити an зовнішній файл, треба використати команду:

```
include  "filename"
```

треба потім замінити filename абсолютним шляхом до файлу з ключем.

Щоб увімкнути сервер host1 необхідно використати ключ для host2 (який має адресу 10.1.7.8 у цьому прикладі), сервер /etc/named.confповинні включити таке правило:

```
server 10.1.7.8 {
   keys { host1- host2.  ;};
 };
```

Аналогічні записи мають бути включені до файлів конфігурації host2.

Далі необхідно додати ключі TSIG для будь-яких ACL, які визначені для IP-адрес і діапазонів адрес до увімкнути безпеку транзакцій. Відповідний запис може виглядати так:

```
allow-update { key host1-host2. ;};
```

Зона, яка вважається безпечною, повинна мати один або кілька ключів зони, пов’язаних із це.

Вони створюються за допомогою dnssec-keygen, як і ключі хосту. Для їх створення використовується алгоритм шифрування DSA ключі.

Згенеровані відкриті ключі повинні бути включені у відповідну зону файл з $INCLUDE.

З командою dnssec-signzone, можна створювати набори згенеровані ключі (keyset-файли), передати їх у файл батьківську зону в безпечний спосіб і підписати їх. Це генерує файли до включити для кожної зони в /etc/named.conf.

2.7 Висновки

В розділі було описано процес проєктування програмно-технічного засобу, який включає:

1 Встановлення DNS-сервер.

2 Конфігурація з YaST.

3 Майстер конфігурації.

4 Налаштування повідомлень про помилки.

5 Редактор зони (NS Records та SOA), а також процес налаштування зон.

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСІБУ КЕРУВАННЯ КОНФІГУРАЦІЄЮ СИСТЕМИ ДОМЕНИХ ІМЕН НА ОСНОВІ BIND

3.1 Налаштування BIND як DNS-сервер приватної мережі на базі OPENSUSE LINUX

Важливим елементом керування конфігурацією та інфраструктурою сервера є підтримка зручного способу перегляду мережевих інтерфейсів та IP-адрес по імені за допомогою налаштування коректної системи доменних імен (DNS).

Використання повних доменних імен (FQDN), а не IP-адреси, для вказівки мережних адрес полегшує налаштування служб та програм і підвищує підтримуваність файлів конфігурації.

Налаштування власної DNS для вашої приватної мережі – це чудовий спосіб удосконалення методів керування серверами.

В роботі здійснено виконано налаштування внутрішнього DNS-сервера за допомогою програмного забезпечення сервера імен BIND (BIND9) на OpenSUSE Leap 15.3, яке може бути використане вашими серверами для надання приватних імен хостів та приватних IP-адрес.

Це служить центральним засобом для управління внутрішніми іменами хостів та приватними IP-адресами, що необхідно при розширенні вашого середовища до більш ніж кількох хостів.

Новий сервер OpenSUSE Leap 15.3, який використовуватиметься як основний DNS-сервер, ns1 .

Другий сервер OpenSUSE Leap 15.3, який використовується як додатковий DNS-сервер, ns2 .

Додаткові сервери в одному центрі обробки даних, які використовуватимуть ваші DNS-сервери.

На кожному з цих серверів необхідно настроїти адміністративний доступ за допомогою користувача sudo та брандмауера з початкового настроювання сервера OpenSUSE Leap 15.3 .

Застосовується два сервери, які будуть призначені як сервери імен DNS ns1 та ns2 .

Є два додаткові клієнтські сервери, які використовуватимуть створену нами інфраструктуру DNS. У роботі було використано назви host1 і host2.

Можна додати будь-яку кількість для інфраструктури.

Всі ці сервери існують у одному центрі обробки даних. Вважатимемо, що це центр обробки даних nyc3 .

Для всіх цих серверів активовано опцію приватної мережі (і в підмережі 10.128.0.0/16;, можливо, потрібно буде редагувати ці параметри для серверів).

Усі сервери запущеного на my mysite.

Оскільки система DNS буде повністю внутрішньою та приватною, не потрібно буде купувати доменне ім'я.

Однак використання власного домену допоможе уникнути конфліктів з доменами з публічною маршрутизацією.

З урахуванням цих припущень буде корисно використовувати схему іменування, яка використовує mysync. mysite для звернення до приватної підмережі або зони.

Таким чином, приватним повним доменним ім'ям (FQDN) для host1 буде mysync. mysite.

Відповідну інформацію див. у наступній таблиці:

Host Role Private FQDN Private IP address

ns1 Basic DNS server ns1.nyc3. mysite 10.128.15.11

ns2 Additional DNS server ns2.nyc3. mysite 10.128.18.12

host1 Standard host 1 host1.nyc3. mysite 10.128.00.11

host2 Standard host 2 host2.nyc3. mysite 10.128.20.12

Примітка: існуюче налаштування буде відрізнятися, приклади імен та IP-адрес будуть використовуватися для демонстрації того, як виконати налаштування DNS-сервера для отримання чинної внутрішньої DNS.

Є можливість адаптувати дане налаштування для середовища, замінивши імена хостів та приватні IP-адреси на власні.

Немає необхідності використовувати ім'я регіону центру обробки даних у схемі присвоєння імен, але було використано його для позначення хостів, що належать до приватної мережі конкретного центру обробки даних.

Якщо використовувати кілька центрів обробки даних, то можна настроїти внутрішню DNS всередині кожного окремого центру обробки даних.

Початок налаштування стартує з установки основного DNS-сервера, ns1.

На обох DNS-серверах, ns1 і ns2 , необхідно оновити кеш пакета apt за допомогою наступної команди:

```
sudo apt-get update
```

Наступним кроком є установка BIND:

```
sudo apt-get install bind bindutils bind-doc
```

Налаштування режиму IPv6 для Bind.

Перш ніж продовжити, необхідно налаштувати режим IPv6 в BIND, оскільки приватна мережа використовує виключно IPv4.

На обох серверах було відредаговано файл налаштувань за замовчуванням bind за допомогою наступної команди:

```
sudo nano /etc/default/bind
```

Було додано “-6” до кінця параметра OPTIONS. Результат виглядатиме так:

```
/etc/default/bind
. .
OPTIONS="-u bind -6"
```

Далі необхідно зберігти файл та закрити його після завершення.

Далі необхідно перезапустити BIND для набуття чинності:

```
sudo systemctl restart bind9
```

Тепер, після установки BIND, необхідно було налаштувати основний DNS-сервер.

3.2 Настройка основного DNS-сервера

Конфігурація BIND складається з файлів, які включені до основного файлу конфігурації, named.conf.

Ці імена файлів починаються з named, тому що це ім'я процесу, який запускає BIND (скорочення від “domain name daemon”).

З цією метою було налаштовано файл параметрів.

3.3 Настройка файлу параметрів

На сервері ns1 було відкрито файл named.conf.options для редагування:

```
sudo nano /etc/bind/named.conf.options
```

Над існуючим блоком optionsстворіть новий блок ACL (список контролю доступу) під назвою "trusted".

Тут було створено список клієнтів, для яких було дозволено рекурсивні DNS-запити (тобто запити від серверів, що знаходяться в тому ж центрі обробки даних, що й ns1).

За допомогою приватних IP-адрес було додано ns1 , ns2 , host1 і host2 в список надійних клієнтів:

```
/etc/bind/named.conf.options — 1 з 3
acl "trusted" {
  10.128.18.10; # ns1 - can be set to localhost
  10.128.20.10; # ns2
  10.128.20.101;  # host1
  10.128.20.12;  # host2
};
options {
  . . .
```

Тепер, коли наявний список довірених DNS-клієнтів, потрібно відредагувати блок options:

```
/etc/bind/named.conf.options — 2 з 3
```

```
  . . .
};
options {
  directory "/var/cache/bind";
  . . .
}
```

Під директивою directory було додано виділені кольором рядки конфігурації (і замінено у відповідній IP-адресі ns1), де результат виглядає так:

```
/etc/bind/named.conf.options — 3 з 3
  . . .
};
options {
  directory "/var/cache/bind";
  recursion yes;   # дозволяє ресурс chueris
  allow-recursion { trusted; };  # дозволяє виконувати рекурсивні запити від «довірених» клієнтів
   listen-on { 10.128.20.101; };   # приватна IP-адреса ns1 - слухає лише в приватні мережі
 allow-transfer { none; };   # вимкнути передачу зон за замовчуванням
  forwarders {
 8.8.8.8;
 8.8.4.4;
  };
  . . .
};
```

Після завершення редагування було збережено та закрито файл named.conf.options.

Відповідно до конфігурації вище, тільки власні сервери (тобто довірені) зможуть запитувати у вашого DNS-сервера зовнішні домени.

3.4 Налаштування локального файлу

Далі необхідним було здійснити налаштування локального файлу, щоб встановити DNS-зони.

На сервері ns1 було відкрито файл named.conf.localдля редагування:

```
sudo nano /etc/bind/named.conf.local
```

Файл має містити лише кілька коментарів. Тут було поставлено зони.

DNS-зони визначають конкретну область для керування та визначення записів DNS.

Оскільки наші домени будуть знаходитися в субдомені mynyc3. mysite, було використано його як зону прямого перегляду.

Оскільки приватні IP-адреси сервера знаходяться у просторі IP-адрес 10.128.0.0/16, було створено зону зворотного перегляду, щоб можна було визначити зворотний перегляд у цьому діапазоні.

Також було додано зону прямого перегляду з наступними рядками, замінивши ім'я зони на власне, та закриту IP-адресу додаткового DNS сервера у директиві allow-transfer:

/etc/bind/named.conf.local — 1 з 2

```
zone "nyc3. mysite" {
 type master;
 file "/etc/bind/zones/db.nyc3. mysite"; # zone file path
 allow-transfer { 10.128.20.12; };  # ns2 private IP address - secondary
};
```

Якщо приватною підмережею є 10.128.0.0/16, необхідно додати зону зворотного перегляду за допомогою наступних рядків:

/etc/bind/named.conf.local — 2 з 2

```
 . . .
};
zone "128.10.in-addr.arpa" {
 type master;
 file "/etc/bind/zones/db.10.128";  # 10.128.0.0/16 підмережа
```

```
allow-transfer { 10.128.20.12; };  # ns2 привітна IP адреса
};
```

Якщо сервери охоплюють кілька приватних підмереж, але знаходяться в одному центрі обробки даних, обов'язково необхідно вказати додаткову зону та файл зони для кожної окремої підмережі.

Після додавання всіх необхідних зон було збережено та закрито файл named.conf.local.

3.5 Створення файлу для зони прямого перегляду

Тепер, коли зони вказано у BIND, потрібно створити відповідні файли для зони прямого та зворотного перегляду.

Файл зони прямого перегляду – це місце, де можна визначати записи DNS для прямого перегляду DNS.

Тобто коли DNS отримує запит імені, наприклад, "myhost1.nyc3. mysite", буде шукати файл зони прямого перегляду для отримання відповідної приватної IP-адреси для host1 .

Таким чином, було створено директорію, в якій знаходяться файли зони. Відповідно до конфігурації named.conf.local, це має бути директорія /etc/bind/zones:

```
sudo mkdir /etc/bind/zones
```

При створенні файлу зони для прямого перегляду редагувався файл зони db.local.

Для цього було скопійовано його в необхідне місце за допомогою наступних команд:

```
sudo cp /etc/bind/db.local /etc/bind/zones/db.nyc3. mysite
```

Тепер потрібно відредагувати файл зони для прямого перегляду:

```
sudo nano /etc/bind/zones/db.nyc3. mysite
```

Спочатку він виглядатиме таким чином:

/etc/bind/zones/db.nyc3. mysite — оригінал

```
$TTL 604800
```

```
@ IN   SOA  localhost. root.localhost. (
   2   ; Serial
    604800   ; очищення
   86400   ; повтор спроби
    2419200   ; завершення
    604800 ) ; кеш  TTL
;
@ IN   NS   localhost.   ; видалити
@ IN   A 127.0.0.1 ; видалити
@ IN   AAAA ::1 ; видалити
```

По-перше, необхідно було відредагувати запис SOA.

Потім замінити перший запис “localhost” на повне доменне ім'я (FQDN) ns1, а потім замінити “root.localhost” на “admin.nyc3. mysite”.

При кожній зміні файлу зони було потрібно збільшувати значення serial, перш ніж перезапускати процес named.

В роботі було збільшено на значення до “3”. Тепер файл має виглядати приблизно так:

```
/etc/bind/zones/db.nyc3. mysite — оновлено 1 з 3
@ IN   SOA  ns1.nyc3. mysite. admin.nyc3. mysite. (
   3   ; Serial
   . . .
```

Далі було видалено три записи наприкінці файлу (після запису SOA).

Наприкінці файлу було додано записи імені сервера з наступними рядками (було замінено імена на власні).

В другому стовпці було вказано, що це записи “NS”.

```
/etc/bind/zones/db.nyc3. mysite — оновлено 2 з 3
. . .
; name servers - NS records
 IN   NS   ns1.nyc3. mysite.
 IN   NS   ns2.nyc3. mysite.
```

А для хостів, які належать до цієї зони.

Це може бути будь-який сервер, ім'я якого буде закінчуватися на my.nyc3. mysite (було замінено імена та приватні IP-адреси).

Використовуючи імена та приватні IP-адреси, було додано записи A для ns1 , ns2 , host1 і host2 таким чином:

```
/etc/bind/zones/db.nyc3. mysite — оновлено 3 з 3
. . .
; name servers - A records
ns1.nyc3. mysite. IN   A 10.128.11.101
ns2.nyc3. mysite. IN   A 10.128.18.102
; 10.128.0.0/16 - A records
host1.nyc3. mysite.  IN   A   10.128.20.101
host2.nyc3. mysite.  IN   A   10.128.20.12
```

Далі було збережено та закрито файл db.nyc3. mysite.

Отриманий в результаті файлу зони для перегляду виглядає таким чином з параметрами:

```
/etc/bind/zones/db.nyc3. mysite — оновлено
$TTL 604800
@ IN   SOA  ns1.nyc3. mysite. admin.nyc3. mysite. (
   3  ; Serial
 604800  ; Refresh
  86400  ; Retry
   2419200  ; Expire
 604800 )   ; Negative Cache TTL
;
; name servers - NS records
  IN   NS   ns1.nyc3. mysite.
  IN   NS   ns2.nyc3. mysite.
; name servers - A records
ns1.nyc3. mysite. IN   A 10.128.20.101
ns2.nyc3. mysite. IN   A 10.128.20.12
; 10.128.0.0/16 - A records
```

```
host1.nyc3. mysite.  IN   A   10.128.100.101
host2.nyc3. mysite.  IN   A   10.128.200.102
```

3.6 Створення файлу (файлів) зони для перегляду

Наступним кроком було здійснено створення файлу (файлів) зони для перегляду.

Файли зони зворотного перегляду служать місцем, де можна визначати PTR записів DNS для перегляду DNS.

Тобто коли DNS отримує запит для IP-адреси, наприклад, "10.128.100.101", вона буде шукати файл (файли) зони для зворотного перегляду, щоб отримати відповідне повне доменне ім'я, в нашому випадку це "host1".mynyc3. mysite”.

ns1 для кожної зони зворотного перегляду, заданої у файлі named.conf.local,потрібно створити файл зони для зворотного перегляду.

При створенні файлу (або файлів) зони для перегляду було використано файл зони db.local.

Було скопійовано його в необхідне місце за допомогою наступних команд (було замінено ім'я файлу призначення, щоб воно відповідало визначенню зони для перегляду):

```
sudo cp /etc/bind/db.127 /etc/bind/zones/db.10.128
```

далі було відредаговано файл зони для зворотного перегляду, який відповідає зоні(-ам), визначеній(-им) у named.conf.local:

```
sudo nano /etc/bind/zones/db.10.128
```

Спочатку він виглядатиме приблизно таким чином:

```
/etc/bind/zones/db.10.128 — оригінал
$TTL 604800
@ IN   SOA  localhost. root.localhost. (
  1   ; Serial
   604800   ; Refresh
  86400   ; Retry
   2419200   ; Expire
```

```
    604800 ) ; Negative Cache TTL
;
@ IN   NS   localhost.   ; delete this line
1.0.0   IN   PTR  localhost.   ; delete this line
```

Як і у випадку з файлом зони для перегляду, потрібно змінити запис SOA і збільшити значення serial.

Він повинен виглядати так:

```
/etc/bind/zones/db.10.128 — оновлено 1 з 3
@ IN   SOA  ns1.nyc3. mysite. admin.nyc3. mysite. (
  3   ; Serial
  . . .
```

Далі було видалено два записи в кінці файлу (після запису SOA).

Наприкінці файлу було додано записи імені сервера з наступними рядками (замініть імена на власні).

Необхідно зауважити, що в другому стовпці файлу вказується, що це записи “NS”.

```
/etc/bind/zones/db.10.128 — оновлено 2 з 3
. . .
; name servers - NS records
  IN   NS   ns1.nyc3. mysite.
  IN   NS   ns2.nyc3. mysite.
```

Потім було додано записи PTR для всіх ваших серверів, чия IP-адреса відповідає підмережі файлу зони, який ви редагуєте.

У проєкті було усі наші хости, оскільки всі вони знаходяться у підмережі 10.128.0.0/16.

Необхідно звернути увагу, що перший стовпець включає два останні байти приватних IP-адрес ваших серверів у зворотному порядку .

Тому обов'язково необхідно було замінити імена та приватні IP-адреси згідно з даними ваших серверів:

```
/etc/bind/zones/db.10.128 — оновлено 3 з 3
. . .
```

```
; PTR Records
11.10   IN   PTR  ns1.nyc3. mysite. ; 10.128.20.101
12.20   IN   PTR  ns2.nyc3. mysite. ; 10.128.20.12
101.100 IN   PTR  host1.nyc3. mysite.  ; 10.128.100.101
102.200 IN   PTR  host2.nyc3. mysite.  ; 10.128.200.102
```

Збережіть і закрийте файл зони для перегляду.

Отриманий в результаті приклад файлу з параметрами зони для перегляду виглядає так:

```
/etc/bind/zones/db.10.128 — оновлено
$TTL 604800
@ IN   SOA  nyc3. mysite. admin.nyc3. mysite. (
   3   ; Serial
    604800   ; Refresh
  86400   ; Retry
    2419200   ; Expire
    604800 ) ; Negative Cache TTL
; name servers
   IN   NS   ns1.nyc3. mysite.
   IN   NS   ns2.nyc3. mysite.
; PTR Records
11.10   IN   PTR  ns1.nyc3. mysite. ; 10.128.20.101
12.20   IN   PTR  ns2.nyc3. mysite. ; 10.128.20.12
101.100 IN   PTR  host1.nyc3. mysite.  ; 10.128.100.101
102.200 IN   PTR  host2.nyc3. mysite.  ; 10.128.200.102
```

3.7 Перевірка синтаксису конфігурації BIND

Наступним кроком після редагування файлів є перевірка файли на наявність помилок.

З цією метою було запущено команду для перевірки синтаксису файлів named.conf*:

```
sudo named-checkconf
```

Якщо у іменованих файлах конфігурації немає помилок у синтаксисі, потрібно повернутися до командного рядка без будь-яких повідомлень про помилки.

Якщо виявлено проблеми з конфігураційними файлами, потрібно переглянути повідомлення про помилку та розділ «Налаштування основного DNS сервера», а потім знову скористатися командою named-checkconf.

Команда named-checkzone може використовуватися для перевірки коректності файлів зони.

Перший аргумент команди вказує ім'я зони, а другий аргумент визначає відповідний файл зони, обидва з яких визначені в named.conf.local.

Наприклад, щоб перевірити конфігурацію зони прямого перегляду “ nyc3. mysite ”, було запущено наступну команду (змінено на імена зони прямого перегляду та файлу):

```
sudo named-checkzone nyc3. mysite db.nyc3. mysite
```

Щоб перевірити конфігурацію зони для перегляду “ 128.10 .in-addr.arpa”, було запущено команду (замінено на дані, що відповідають зоні для перегляду та файлу):

```
sudo named-checkzone 128.10.in-addr.arpa /etc/bind/zones/db.10.128
```

3.8Перезапуск BIND

Коли всі файли конфігурації та зони не мають помилок, було здійснено перезапуск служби BIND.

Перезапуск здійснено таким чином BIND:

```
sudo systemctl restart bind9
```

Якщо є брандмауер UFW, то необхідно відкрити доступ до BIND за допомогою команди:

```
sudo ufw allow Bind9
```

Тепер основний сервер DNS налаштований і може відповідати на запити DNS.

Наступним кроком є створення додаткового DNS-сервера.

3.9 Налаштування додаткового DNS-сервера

З метою забезпечення безперебійного функціонування мережі правильним рішенням є створення додаткового DNS-сервера, який відповідатиме на запити, якщо основний сервер виявиться недоступним.

Таким чином, було здійснено налаштування додаткового сервера.

На сервері ns2 було відредаговано файл named.conf.options:

```
sudo nano /etc/bind/named.conf.options
```

У верхній частині файлу було додано ACL із приватними IP-адресами всіх ваших довірених серверів:

```
/etc/bind/named.conf.options — оновлено 1 з 2 (вторинний)
acl "trusted" {
  10.128.20.101;   # ns1
  10.128.20.12;   # ns2 - can be set to localhost
  10.128.100.101;  # host1
  10.128.200.102;  # host2
};
options {
  . . .
```

Під директивою directory було додано наступні рядки:

```
/etc/bind/named.conf.options — оновлено 2 з 2 (вторинний)
  recursion yes;
  allow-recursion { trusted; };
  listen-on { 10.128.20.12; };   # ns2 private IP address
  allow-transfer { none; }; # disable zone transfers by default
  forwarders {
 8.8.8.8;
 8.8.4.4;
  };
```

Потім було збережено та закрито файл named.conf.options.

Цей файл має виглядати так само, як файл named.conf.optionscервера ns1 , за винятком того, що його необхідно налаштувати на прослуховування приватної IP-адреси ns2 .

Тепер потрібно відредагувати файл named.conf.local:

```
sudo nano /etc/bind/named.conf.local
```

Було визначено slave-зони, що відповідають master-зонам основного DNS-сервера.

Необхідно звернути увагу, що як тип використовується slave, у файлі відсутня шлях, й існує директива masters, яка має бути налаштована на приватну IP-адресу основного DNS-сервера.

Якщо було визначено кілька зон для зворотного перегляду на основному DNS-сервері, обов'язково перевірте, чи всі вони були додані на цьому етапі:

```
/etc/bind/named.conf.local — оновлено (вторинний)
zone "nyc3. mysite" {
 type slave;
 file "db.nyc3. mysite";
 masters { 10.128.20.101; };  # ns1 private IP
};
zone "128.10.in-addr.arpa" {
 type slave;
 file "db.10.128";
 masters { 10.128.20.101; };  # ns1 private IP
};
```

Потім було збережено та закрито файл named.conf.local.

Далі було запущено наступну команду для перевірки валідності файлів конфігурації:

```
sudo named-checkconf
```

Після перевірки було перезапущено BIND:

```
sudo systemctl restart bind9
```

Наступним кроком було надано дозвіл підключення DNS до сервера, змінивши правила брандмауера UFW:

```
sudo ufw allow Bind9
```

Тепер конфігурація має основний та додатковий DNS-сервери для імені приватної мережі та перетворення IP-адреси.

3.10 Налаштування DNS-клієнтів

Наступним кроком було налаштування клієнтських серверів, щоб вони могли використовувати приватні DNS-сервери.

Перш ніж усі сервери в довіреному ACL зможуть надсилати запити на ваші DNS-сервери, необхідно налаштувати для кожного з них використання ns1 та ns2 як сервер імен.

Цей процес варіюється в залежності від операційної системи, але для більшості дистрибутивів Linux він має на увазі додавання ваших серверів доменних імен до файлу /etc/resolv.conf.

3.11 Клієнти OpenSUSE Leap 15.3

На OpenSUSE Leap 15.3 налаштування мережевої взаємодії виконується за допомогою Netplan, абстракції, яка дозволяє записувати стандартну конфігурацію мережі та застосовувати її до несумісного мережного ПЗ, що відповідає за використовуваний бекенд.

Для налаштування DNS нам потрібно записати файл конфігурації служби Netplan.

З цією метою було приєднано пристрій, пов'язаний з приватною мережею, надіславши приватній підмережі команду ip address:

```
ip address show to 10.128.0.0/16
```

Output

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
```

```
inet 10.128.100.101/16 brd 10.128.255.255 scope global eth1
 valid_lft forever preferred_lft forever
```

У роботі було використано приватний інтерфейс eth1.

Далі необхідно було створити новий файл у /etc/netplanз ім'ям 00-private-nameservers.yaml:

```
sudo nano /etc/netplan/00-private-nameservers.yaml
```

було відредаговано файл наступний вміст.

Потрібно було змінити інтерфейс приватної мережі, адреси ваших DNS-серверів ns1 і ns2 , а також зону DNS:

Примітка: Netplan використовує формат серіалізації даних YAML для файлів конфігурації:

```
/etc/netplan 00-private-nameservers.yaml
network:
 version: 2
 ethernets:
  eth1:   # Private network interface
   nameservers:
 addresses:
 - 10.128.20.101 # Private IP for ns1
 - 10.132.20.12 # Private IP for ns2
 search: [ nyc3. mysite ]  # DNS zone
```

далі було збережено файл та закрито його після завершення.

Потім потрібно було повідомити Netplan про необхідність використання нового конфігураційного файлу за допомогою команди:

```
netplan try.
```

За наявності проблем, що призводять до втрати підключення до мережі, Netplan автоматично перезапускатиме зміни після закінчення певного періоду часу:

```
 sudo netplan try
```

Output

Warning: Stopping systemd-networkd.service, but it can still be activated by:

systemd-networkd.socket

Do you want to keep these settings?

Press ENTER before the timeout to accept the new configuration

Changes will revert in 120 seconds

Якщо лічильник у нижній частині оновлюється коректно, це означає, що нові конфігурації вдалося принаймні не пошкодити ваше з'єднання SSH.

Після натиснення ENTER було змінено конфігурацію.

Далі було перевірено DNS-перетворювач системи, щоб визначити, чи застосовані зміни до конфігурації DNS:

sudo systemd-resolve --status

В розділі для інтерфейсу приватної мережі вказано приватні IP-адреси ваших DNS-серверів, які будуть перераховані насамперед, а за ними йдуть резервні значення.

Конфігурація домену знаходитись у рядку DNS Domain:

Output

. . .

Link 3 (eth1)

Current Scopes: DNS

LLMNR setting: yes

MulticastDNS setting: no

DNSSEC setting: no

DNSSEC supported: no

DNS Servers: 10.128.20.101

10.128.20.12

67.207.67.2

67.207.67.3

DNS Domain: nyc3. mysite

. . .

Клієнт тепер налаштований на використання для використовуваних внутрішніх DNS-серверів.

У серверах Ubuntu 16.04 та Debian можна змінити файл /etc/network/interfaces:

```
sudo nano /etc/network/interfaces
```

В рядок dns-nameservers було додано сервери доменних імен на початок списку, який вже доданий до файлу.

Під цим рядком було додано опцію dns-search, що вказує на базовий домен інфраструктури.

У роботі це myserver/mysite:

```
/etc/network/interfaces
. . .
dns-nameservers 10.128.20.101 10.128.20.12 8.8.8.8
dns-search nyc3. mysite
. . .
```

Далі було збережено файл та закрито його після завершення.

Потім було перезапущено мережеві служби, застосовуючи зміни за допомогою наступних команд.

Щоб переконатися, що було замінено eth0на ім'я мережевого інтерфейсу було використано команду:

```
sudo ifdown --force eth0 && sudo ip addr flush dev eth0 && sudo ifup --force eth0
```

В результаті було виконано перезапуск мережі без відключення поточного підключення.

Щоб побачити, що все працює коректно, необхідно отримати:

```
Output
RTNETLINK answers: No such process
Waiting for DAD... Done
```

Для перевірки налаштування щодо їх застосовності, було використано команду:

```
cat /etc/resolv.conf
```

як результат було отримано інформацію про сервери доменних імен у файлі /etc/resolv.conf, а також домен пошуку:

Output

Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)

DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN

nameserver 10.128.20.101

nameserver 10.128.20.12

nameserver 8.8.8.8

search nyc3. mysite

Таким чином ,було отримано клієнт, налаштований для використання DNS-серверів.

Клієнти CentOS. У CentOS, RedHat та Fedora Linux необхідно було відредагувати файл /etc/sysconfig/network-scripts/ifcfg-eth0.

Також було замінити eth0на ім'я основного мережного інтерфейсу:

sudo nano /etc/sysconfig/network-scripts/ifcfg-eth0

для цього було знайдено опції DNS1і DNS2 та встановлено для них приватні IP-адреси ваших основного та додаткового серверів доменних імен.

Потім було додано параметр DOMAIN за допомогою базового домену інфраструктури.

У роботі це було “nyc3. mysite”:

/etc/sysconfig/network-scripts/ifcfg-eth0

. . .

DNS1=10.128.20.101

DNS2=10.128.20.12

DOMAIN='nyc3. mysite'

. . .

Далі було збережено та закрито його після завершення.

Потім було перезапущено мережу за допомогою команди:

sudo systemctl restart network

Команда може не відповідати кілька секунд, але через короткий час стає можливим повернутися до командного рядка.

Що переконатися, що зміни набули чинності, було введено команду:

cat /etc/resolv.conf

Таким чином, було отримано видимість сервери доменних імен та домену пошуку у списку:

/etc/resolv.conf

nameserver 10.128.20.101

nameserver 10.128.20.12

search nyc3. mysite

3.12 Тестування клієнтів

Після виконаних вище налаштувань тепер може підключитися клієнт та використовувати DNS-сервери.

Щоб перевірити, для перевірки того, чи ваші клієнти можуть надсилати запити вашим серверам доменних імен, було використано команду nslookup.

Існує можливість зробити це для всіх клієнтів, які були налаштовані і знаходяться в довіреному ACL.

Для клієнтів CentOS може знадобитися встановлення утиліти за допомогою наступної команди:

sudo yum install bind-utils

Можна почати прямий перегляд.

Прямий перегляд. Наприклад, можна виконати прямий перегляд для отримання IP-адреси host1.nyc3. mysite за допомогою наступної команди:

nslookup host1

Запит "host1" розширюється до "host1.nyc3. mysite", тому що опція searchзаданий для приватного субдомену, а запити DNS намагатимуться переглянути цей субдомен перед пошуком по всьому хосту.

Результат описаної вище команди виглядав так:

Output

Server: 127.0.0.53

Address: 127.0.0.53#53

Non-authoritative answer:

Name: host1.nyc3. mysite

Address: 10.128.100.101

Тепер можна перевірити зворотний перегляд.

Зворотній перегляд. Щоб протестувати зворотний перегляд, було надіслано DNS-серверу запит на приватну IP-адресу host1 :

nslookup 10.128.100.101

Результат виглядав так:

Output

11.10.128.10.in-addr.arpa   name = host1.nyc3. mysite.

Authoritative answers can be found from:

Якщо всі імена та IP-адреси було передано правильні значення, це означало, що файли зони налаштовані належним чином.

Якщо було отримано несподівані значення, обов'язково потрібно перевірити файли зони на основному DNS-сервері (наприклад, db.nyc3. mysitei db.10.128).

3.13 Збереження DNS-записів

Оскільки внутрішні DNS-сервери налаштовані належним чином, тепер є можливим перейти до збереження записів зони.

Тепер, коли є внутрішній DNS-сервер, потрібно зберігати записи DNS, щоб вони точно відображали середовище сервера.

Додавання хоста до DNS. При додаванні хоста до середовища (в одному центрі обробки даних) потрібно додати його до DNS.

Для цього було виконати кілька кроків:

Основний сервер доменних імен. Файл зони для перегляду: додайте запис A для нового хоста, збільшивши значення “Serial”.

Файл зони для перегляду: додайте запис PTR для нового хоста, збільшивши значення “Serial”.

Додати приватну IP-адресу нового хоста в довірений ACL (named.conf.options).

Протестувати ваші файли конфігурації:

```
sudo named-checkconf
sudo named-checkzone nyc3. mysite db.nyc3. mysite
sudo named-checkzone 128.10.in-addr.arpa /etc/bind/zones/db.10.128
```

Потім перезавантажити BIND:

```
sudo systemctl reload bind9
```

Основний сервер має бути налаштований для використання нового хоста.

Додатковий сервер доменних імен. Було додано приватну IP-адресу нового хоста в довірений ACL (named.conf.options)

Потім було перевірено синтаксис конфігурації:

```
sudo named-checkconf
```

Потім було перезавантажено BIND:

```
sudo systemctl reload bind9
```

Вторинний сервер тепер здатний приймати підключення з нового хоста.

Потім було налаштовано новий хоста для використання DNS, а також налаштовано /etc/resolv.confдля використання ваших DNS-серверів.

Також було виконано перевірку за допомогою команди:

```
nslookup.
```

3.14 Видалення хоста з DNS

Якщо є виконати видалення хосту з середовища або прибрати його з DNS, необхідно видаліть всі дані, які були додані при додаванні сервера до DNS (тобто виконати описані вище кроки у порядку).

Тепер можна звертатися до інтерфейсів серверів приватної мережі на ім'я, а не на IP-адресу.

Це спрощує налаштування служб та програм, оскільки більше не потрібно запам'ятовувати приватні IP-адреси, а файли легше читати і розуміти.

Крім того, тепер можна змінювати конфігурації для роботи з новими серверами в одному місці, на вашому основному DNS-сервері, замість того, щоб редагувати цілий набір різних файлів.

Коли внутрішній DNS-сервер був налаштований, а файли конфігурації використовують приватні FQDN для вказівки мережних підключень, критично важливо, щоб ваші DNS-сервера обслуговувалися належним чином.

Якщо обидва сервери будуть недоступні, служби та програми, які спираються на них під час роботи, не зможуть нормально функціонувати.

Саме тому рекомендується налаштувати для вашої DNS мінімум один додатковий сервер та зберігати робочі резервні копії всіх серверів.

3.15 Висновок

В розділі було описано програмно-апаратна реалізацію та тестування програмно-технічного засобу керування конфігурацією системи домених імен на основі BIND, який включає: налаштування BIND як DNS-сервер приватної мережі на базі OPENSUSE LINUX, настройка основного DNS-сервера, настройка файлу параметрів, налаштування локального файлу, створення файлу для зони прямого перегляду, створення файлу (файлів) зони для перегляду, перевірка синтаксису конфігурації BIND, перезапуск BIND, налаштування додаткового DNS-сервера, налаштування DNS-клієнтів, налаштування клієнтів в OpenSUSE Leap 15.3, тестування клієнтів, збереження DNS-записів, видалення хоста з DNS.

ВИСНОВКИ

В результаті роботи було розроблено програмно-технічний засіб керування конфігурацією системи доменних імен на основі BIND.

В першому розділі представлено основні аспекти функціонування програмного забезпечення для взаємодії з системою доменних імен.

Також в розділі описано основні компоненти BIND як сервісу, що дозволяє реалізувати програмно-апаратна реалізація та тестування програмно-технічного засобу керування конфігурацією системи домених імен на основі BIND.

В другому розділі було описано процес проєктування програмно-технічного засобу, який включає:

1 Встановлення DNS-сервер.

2 Конфігурація з YaST.

3 Майстер конфігурації.

4 Налаштування повідомлень про помилки.

5 Редактор зони (NS Records та SOA), а також процес налаштування зон.

В розділі було описано програмно-апаратна реалізацію та тестування програмно-технічного засобу керування конфігурацією системи домених імен на основі BIND, який включає: налаштування BIND як DNS-сервер приватної мережі на базі OPENSUSE LINUX, настройка основного DNS-сервера, настройка файлу параметрів, налаштування локального файлу, створення файлу для зони прямого перегляду, створення файлу (файлів) зони для перегляду, перевірка синтаксису конфігурації BIND, перезапуск BIND, налаштування додаткового DNS-сервера, налаштування DNS-клієнтів, налаштування клієнтів в OpenSUSE Leap 15.3, тестування клієнтів, збереження DNS-записів, видалення хоста з DNS.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Комп’ютерні системи паралельної обробки даних. Навчальний посібник для студентів напряму підготовки Комп’ютерна інженерія", В.О. Бойчук, О.В. Огнєвий, Ю.В. Хмельницький. Хмельницький: ХНУ, 2011. 250 с.
2. Комп’ютерні системи. Методичні вказівки до виконання лабораторних робіт /О.В.Огнєвий,В.О.Бойчук- Хмельницький : ХНУ, 2009.-94с.
3. Тарарака В.Д. Архітектура комп’ютерних систем: навч. посіб. / В.Д. Тарарака. Житомир: ЖДТУ, 2018. 383 с.
4. Микитишин А.Г., Митник М.М., Стухляк П.Д. Комп’ютерні мережі, Книга 1. Навчальний посібник для технічних спеціальностей ВНЗ. Магнолія 2006, 2018. – 256с.
5. S. S. H. Tse, Online Bounds on Balancing Two Independent Criteria with Replication and Reallocation, *IEEE Transactions on Computers*, vol. 61, no. 11, pp. 1601-1610, Nov. 2012, doi: 10.1109/TC.2011.168.
6. Y. Chen, S. Radhakrishnan, S. K. Dhall and S. Karabuk, On the game server network selection with delay and delay variation constraints, *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, 2011, pp. 1-10, doi: 10.1109/COMSNETS.2011.5716473.
7. D. Li and J. Wu, On Data Center Network Architectures for Interconnecting Dual-Port Servers, *IEEE Transactions on Computers*, vol. 64, no. 11, pp. 3210-3222, 1 Nov. 2015, doi: 10.1109/TC.2015.2389847.
8. D. Ramalingam, Practicing computer hardware configuration and network installation in a virtual laboratory environment: A case study, *2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007*, pp. F3G-21-F3G-24, doi: 10.1109/FIE.2007.4417940.
9. S. Neelakantan et al., An architecture for self-configuration of network for QoS and security, *2009 First International Communication Systems and Networks and Workshops*, 2009, pp. 1-5, doi: 10.1109/COMSNETS.2009.4808843.

10. Комп’ютерні мережі : навчальний посібник / Азаров О. Д., Захарченко С. М., Кадук О. В. та ін. Вінниця : ВНТУ, 2013. 371с.

11. Буров Є. В. Комп'ютерні мережі: підручник / Євген Вікторович Буров. Львів: «Магнолія 2006», 2010. 262 с.

12. Кулаков Ю.О., Жуков І.А. Комп’ютерні мережі. Навчальний посібник/ за ред. Кулакова Ю.О. К: НАУ, 2009. -392 с.

13. А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник. Комп'ютерні мережі Львів: «Магнолія 2006», 2013. 256 с.

14. Лосев Ю. І. Комп'ютерні мережі: навчальний посібник / Ю. І. Лосев, К. М. Руккас,. С. І. Шматков / За редакцією Ю. І. Лосева. Х. : ХНУ імені В. Н. Каразіна, 2013. 248 с.

15. Ситник В.Ф., Козак І.А. Телекомунікації в бізнесі: Навч.-посібник. К.: КНЕУ, 2003. 204с.

16. Валецька Т.М. Комп’ютерні мережі: Апаратні засоби. Навч. посібник.-К.:Центр навч. Літератури, 2002.

17. Лозікова Г.М. Комп’ютерні мережі.Навч.-методичний посібник.-К.:Центр навч. Літератури, 2004.

18. Спартак Марк, Паппас Френк и др. Компьютерные сети и сетевие технологии: Пер. с англ. К.:000 “ТНД ДС”, 2002. 736 с.

19. Веттіі Дітер. Novell NetWare. К.: Торгово-видавниче бюро BHV, 2013.

20. Гольц Г. Робочі станції і інформаційні мережі. М.: Машинобудування, 2000.

21. Жуков Ігор Анатолійович. Комп`ютерні мережі та технології: навчальний посібник /Жуков І.А., Гуменюк В.О., Альтман І.Є./ К.: НАУ, 2004.-276с.

22. Кулаков Ю. А., Луцкий Г. М. Компьютерные сети. К.: Юниор, 2012. 384 с.

23. Нессер Д. Дж. Оптимизация и поиск неисправностей в сетях. Киев: Диалектика, 2006.384 с.

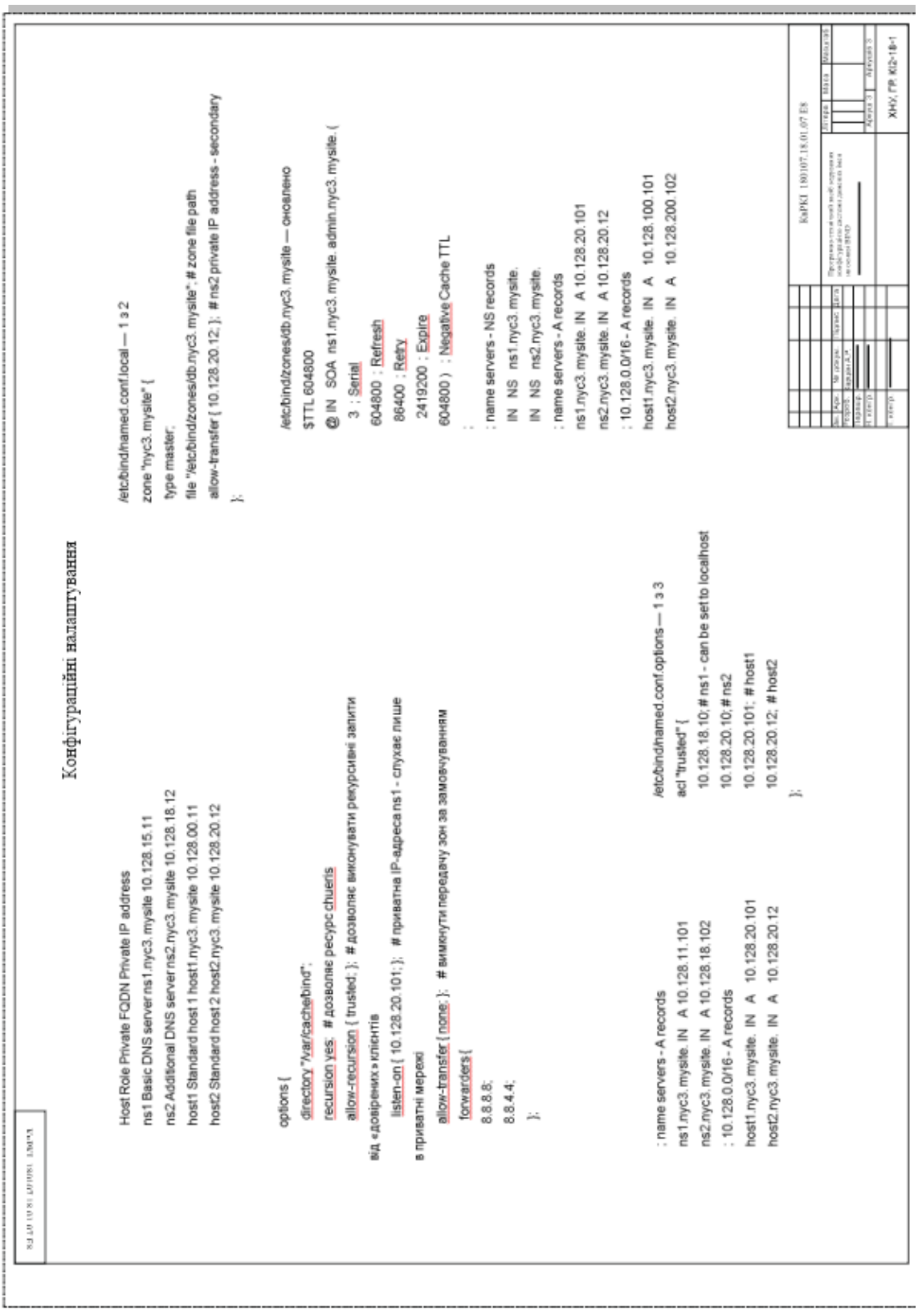
24. Швиденко М.З., Матус Ю.В.. Технології комп’ютерних мереж. Навч.-метод. посібник., Київ. Видавництво ООО “Береста”, 2007.

25. Anylogic. URL: https://www.anylogic.com/.
26. OpenSUSE . URL: https://software.opensuse.org/ .
27. Ubuntu Server 20.04 LTS. URL: https://ubuntu.com/download/server.
28. Windows 10. URL: www.microsoft.com.

Додаток А

(обов’язковий)

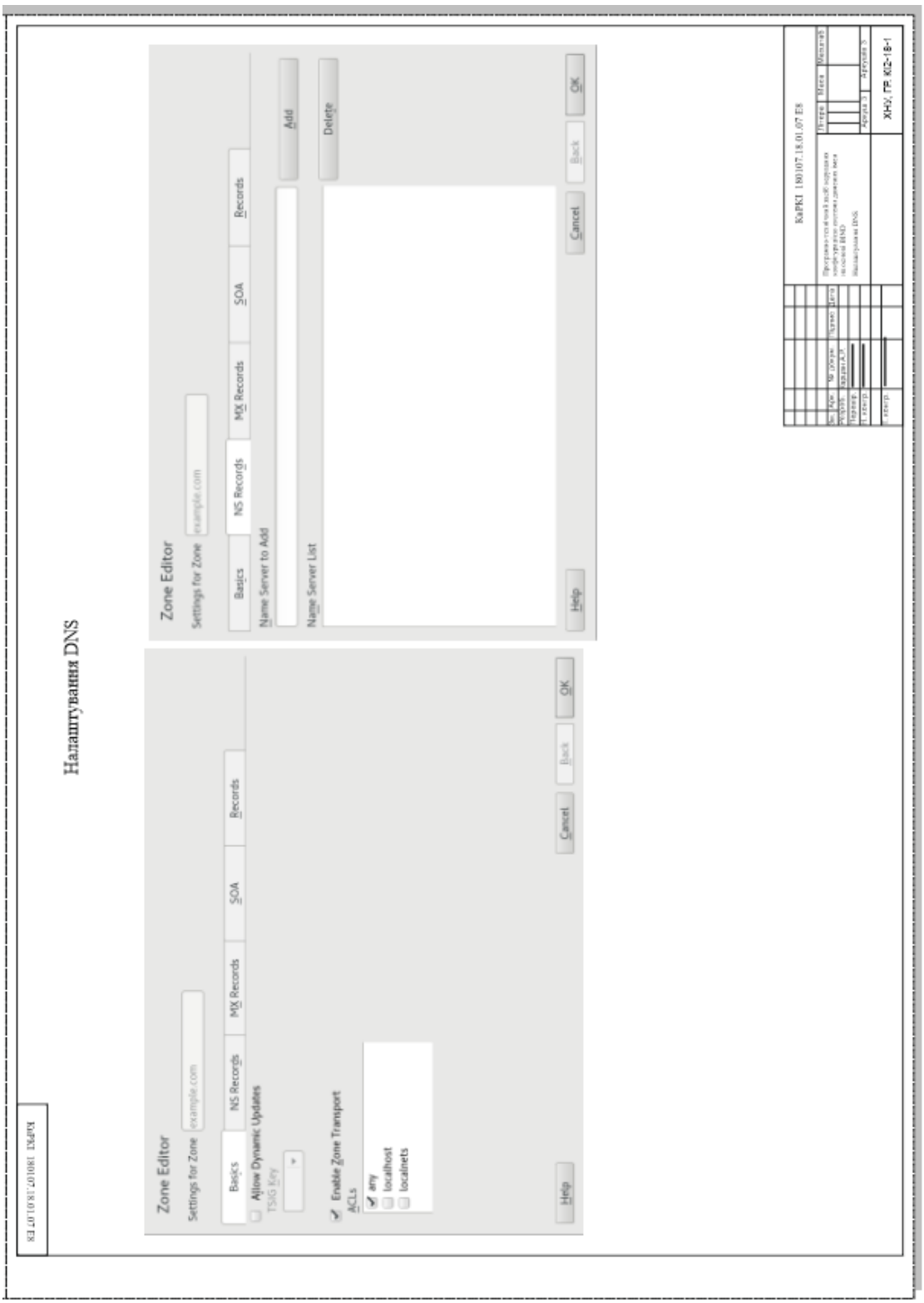
Копія креслення «Конфігураційні налаштування»



Додаток Б

(обов’язковий)

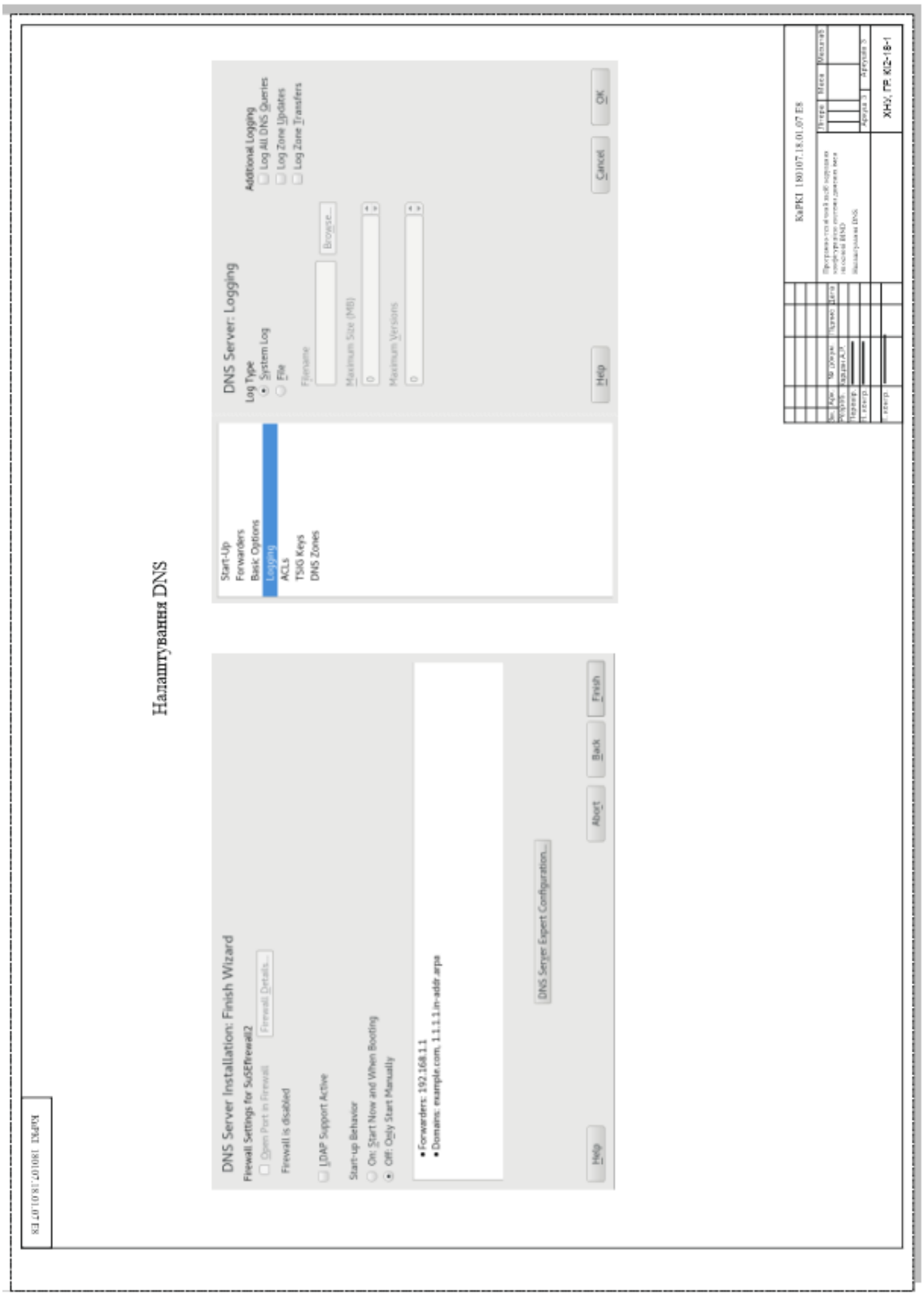
Копія креслення «Налаштування DNS»



Додаток В

(обов’язковий)

Копія креслення «Налаштування DNS»



UNICHECK

Ім'я користувача:
Кафедра КІ

ID перевірки:
1011337354

Дата перевірки:
25.05.2022 16:47:58 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
25.05.2022 16:49:44 EEST

ID користувача:
100005591

Назва документа: **Карцан_Програмно-технічний засіб керування конфігурацією та інфраструктурою локальн...**

Кількість сторінок: **59** Кількість слів: **10400** Кількість символів: **71903** Розмір файлу: **605.50 KB** ID файлу: **1011223018**

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

8.1%
Схожість

Найбільша схожість: **5.2%** з Інтернет-джерелом (https://www.xpresservers.com/%D0%BD%D0%B0%D1%81%D1%82%D1%

7.03% Джерела з Інтернету	104	Сторінка 61
1.17% Джерела з Бібліотеки	97	Сторінка 61

0% Цитат

Не знайдено жодних цитат

Не знайдено жодних посилань

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	93
Підозріле форматування	14 сторінок

Wed May 25 15:59:27 EEST 2022, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 11%**

	Документ		Суммарное совпадение по Базе Данных	
ID: 103976 Название: Програмно-технічний засіб керування конфігурацією та інфраструктурою локального центру обробки даних Добавлено в БД: 2022-05-25 Авторы: А.Р. Карцан Руководители: С.М. Лисенко Консультанты: Опоненты:	Символы	Лексемы	Символы	Лексемы
	59970	614	1460 (2%)	19 (3%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Карцан Артур Русланович

Тема: Програмно-технічний засіб керування конфігурацією системи домених імен на основі BIND

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень ___3____ Кількість сторінок записки ___58___

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка програмно-технічного засобу керування конфігурацією системи домених імен на основі BIND.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі представлено основні аспекти функціонування програмного забезпечення для взаємодії з системою доменних імен.

Також в розділі описано основні компоненти BIND як сервісу, що дозволяє реалізувати програмно-апаратна реалізація та тестування програмно-технічного засобу керування конфігурацією системи домених імен на основі BIND.

В другому розділі було описано процес проєктування програмно-технічного засобу, який включає:

1 Встановлення DNS-сервер.

2 Конфігурація з YaST.

3 Майстер конфігурації.

4 Налаштування повідомлень про помилки.

5 Редактор зони (NS Records та SOA), а також процес налаштування зон.

В розділі було описано програмно-апаратна реалізацію та тестування програмно-технічного засобу керування конфігурацією системи домених імен на основі BIND,

який включає: налаштування BIND як DNS-сервер приватної мережі на базі OPENSUSE LINUX, настройка основного DNS-сервера, настройка файлу параметрів, налаштування локального файлу, створення файлу для зони прямого перегляду, створення файлу (файлів) зони для перегляду, перевірка синтаксису конфігурації BIND, перезапуск BIND, налаштування додаткового DNS-сервера, налаштування DNS-клієнтів, налаштування клієнтів в OpenSUSE Leap 15.3, тестування клієнтів, збереження DNS-записів, видалення хоста з DNS.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага рішенням на платформі Windows.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре 4.0 / С

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Джулій В.М., к.т.н., доцент, кафедри кібербезпеки Хмельницького національного університету _____

"___"_____2022 р. _____(підпис)

Завідувачу кафедри КІСП
д-ру техн.наук, проф. Говорущенко Т. О.

Карцан А.Р.
ПІБ здобувача вищої освіти

ФПКТС, 4 курсу, групи КІ-18-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

_____
дата

підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічний засіб керування конфігурацією системи домених імен на основі BIND

Автор: Карцан Артур Русланович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Лисенко Сергій Миколайович, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділі аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів щодо використаних програмних скриптів, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 8.1% і адресується до 104 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру практичної роботи і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ С.М. Лисенко

Гарант ОП _____ С. М. Лисенко

Завідувач кафедри КІСП _____ Т. О. Говорущенко