Форма № Н-9.02

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
(повне найменування закладу вищої освіти)

Навчально-науковий інститут інформаційних технологій і робототехніки
(повне найменування інституту, назва факультету (відділення))

Кафедра автоматики, електроніки та телекомунікацій
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

магістр
(ступінь вищої освіти)

на тему **Дослідження принципів функціонування системи доменних імен**

Виконав: студент 6 курсу, групи 601дТТ
спеціальності 172 «Телекомунікації та Радіотехніка»
(шифр і назва напряму підготовки, спеціальності)

Васєв Я.Д.
(прізвище та ініціали)

Керівник Жученко О.С.
(прізвище та ініціали)

Рецензент Штомпель М.А.
(прізвище та ініціали)

Полтава - 2024 рік

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
Інститут Навчально-науковий інститут інформаційних технологій і робототехніки
Кафедра Автоматики, електроніки та телекомунікацій
Ступінь вищої освіти Магістр
Спеціальність 172 «Телекомунікації та радіотехніка»

ЗАТВЕРДЖУЮ

Завідувач кафедри автоматики, електроніки та телекомунікацій
_____О.В. Шефер
"04" 09 202_ р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Васєв Ярослав Дмитрович

1. Тема проекту (роботи) **«Дослідження принципів функціонування системи доменних імен»**
керівник проекту (роботи) Жученко Олександр Сергійович, д.т.н., доцент
затверджена наказом вищого навчального закладу від "4"09 2023 року №986-9А
Вихідні дані до проекту:
Ім'я первинного DNS-сервера: Server; Інтервал оновлень: 50000; Інтервал повторних спроб: 3600; Інтервал закінчення дії: 1209600, Серійний номер: 20231127001
2. Строк подання студентом проекту (роботи) 13.12.2023 р.
3. Дослідження принципів побудови та функціонування системи доменних імен. Дослідження процедур обміну повідомлень DNS, аналіз протоколів. Дослідження роботи DNS сервера створеного на віртуальній машині. Загальні рішення методів періодизації трафіку Wireshark. Розрахунки параметрів DNS-серверів та запитів до них. Розрахунок часу, необхідного для отримання відповіді DNS.
4. Графічна частина: Актуальність роботи, мета роботи, об'єкт дослідження, предмет дослідження; Принципи побудови протоколу DNS; Основи DNS; Формат повідомлення DNS; Принципи функціонування DNS. Проблеми функціонування DNS-служби; Значення мережевих аналізаторів та їх особливості; Загальне використання мережевих аналізаторів; Wireshark; Tcpdump; Висновки.

5. Дата видачі завдання 02.10.2023 р.

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів магістерської роботи	Термін виконання етапів роботи			Примітка (плакати)
1	Аналіз вимог до роботи	05.10.23	I	5%	Пл. 1,2
2	Принципи побудови протоколу DNS	11.10.23		15%	Пл. 3
3	Основи DNS	18.10.23		30%	Пл. 4
4	Формат повідомлення DNS	25.10.23		40%	Пл. 5
5	Принципи функціонування DNS	14.11.23		50 %	Пл. 6
6	Проблеми функціонування DNS-служби	21.11.23	II	60%	Пл. 7
7	Значення мережевих аналізаторів та їх особливості	28.11.23		70%	Пл. 8
8	Загальне використання мережевих аналізаторів	13.12.23	III	100%	Пл. 9,10,11

Магістрант _____ Васєв Я.Д.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Жученко О.С.
(підпис) (прізвище та ініціали)

ЗМІСТ

Вступ

Система доменних імен є ключовим елементом Інтернету, що визначається набором принципів побудови та функціонування. Доменні імена слугують аналогією до поштових адрес, за допомогою якої користувачі можуть легко знаходити та доступатися до веб-ресурсів. У цьому контексті розглядається структура системи доменних імен, принципи її роботи та роль, яку вона відіграє в роботі Інтернету. Актуальність роботи полягає у тому, що система доменних імен є дуже актуальною темою в сучасному інформаційному суспільстві, відіграє ключову роль в Інтернеті, забезпечуючи перетворення легкозапам'ятовуваних доменних імен на IP-адреси, що дозволяє комп'ютерам знаходити один одного в мережі. Дослідження в цій області може принести користь для розуміння та покращення ефективності та безпеки інтернет-інфраструктури. Також, з урахуванням постійного розвитку технологій та збільшення кількості проблем безпеки в Інтернеті, вивчення системи доменних імен залишається актуальним завданням.

Метою роботи є розробка рекомендацій для покращення ефективності, безпеки та інновацій в цій важливій складовій інтернет-інфраструктури.

Для виконання поставленої мети в роботи необхідно виконати наступні завдання:

- розібратися у принципах побудови та функціонування DNS;
- дослідити роботу сервера створеного на віртуальній машині;
- розрахувати параметри серверу та запитів до нього.

Ця робота є результатом глибоких досліджень, етапів аналізу та розробки, проведених мною особисто, з метою дослідження принципів побудови та функціонування DNS, роботи сервера створеного на віртуальній

машині та практики у розрахунках параметрів параметрів серверу та запитів до нього.

РОЗДІЛ 1
ДОСЛІДЖЕННЯ ПРИНЦИПІВ ПОБУДОВИ ТА ФУНКЦІОНУВАННЯ DNS

1.1 Принципи побудови протоколу DNS

Для отримання відповідності між іменами вузлів та їх IP - адресами використовується протокол DNS (Domain Name Service). Система доменних імен являє собою розподілену базу даних, що використовуються застосуванням TCP/IP для встановлення даної відповідальності. Також DNS використовується для маршрутизації електронної пошти. Термін «розподілена база» даних означає, що вона зберігається не на одному мережевому вузлі. Кожен вузол підтримує власну інформаційну базу даних, а також запускає застосування, що відправляє запити до інших серверів. Протокол DNS дозволяє клієнтам і серверам спілкуватися між собою.

Визначник (resolver) виконує доступ до DNS застосування. Визначник – це підпрограма, що використовується для створення відправлення й інтерпретації пакетів, що використовуються серверами імен у мережі. Для мережевої програми потрібно виконати перетворення імені вузла у IP-адресу перед тим, як вона почне відкривати TCP-з’єднання, або відправляти дейтаграму з використанням UDP. Концепції DNS описано у RFC 1034, а деталі розробки та специфікації DNS викладено у RFC 1035.

1.1.1 Основи DNS

Деревоподібну ієрархічну структуру має простір DNS імен. На рисунку 1.1 показано організацію DNS.

Мітку довжиною 63 символи має кожен вузол на рисунку 1.1. Корінь дерева – це спеціальний вузол без мітки. Мітки можуть містити літери

верхнього або нижнього реєстрів. Ім’я домену для будь-якого вузла в дереві – це послідовність міток, що починається з кореневого вузла. При цьому мітки розділяють крапками. Унікальне ім’я домену має мати кожен вузол дерева, але в різних точках дерева можуть використовуватися однакові мітки.

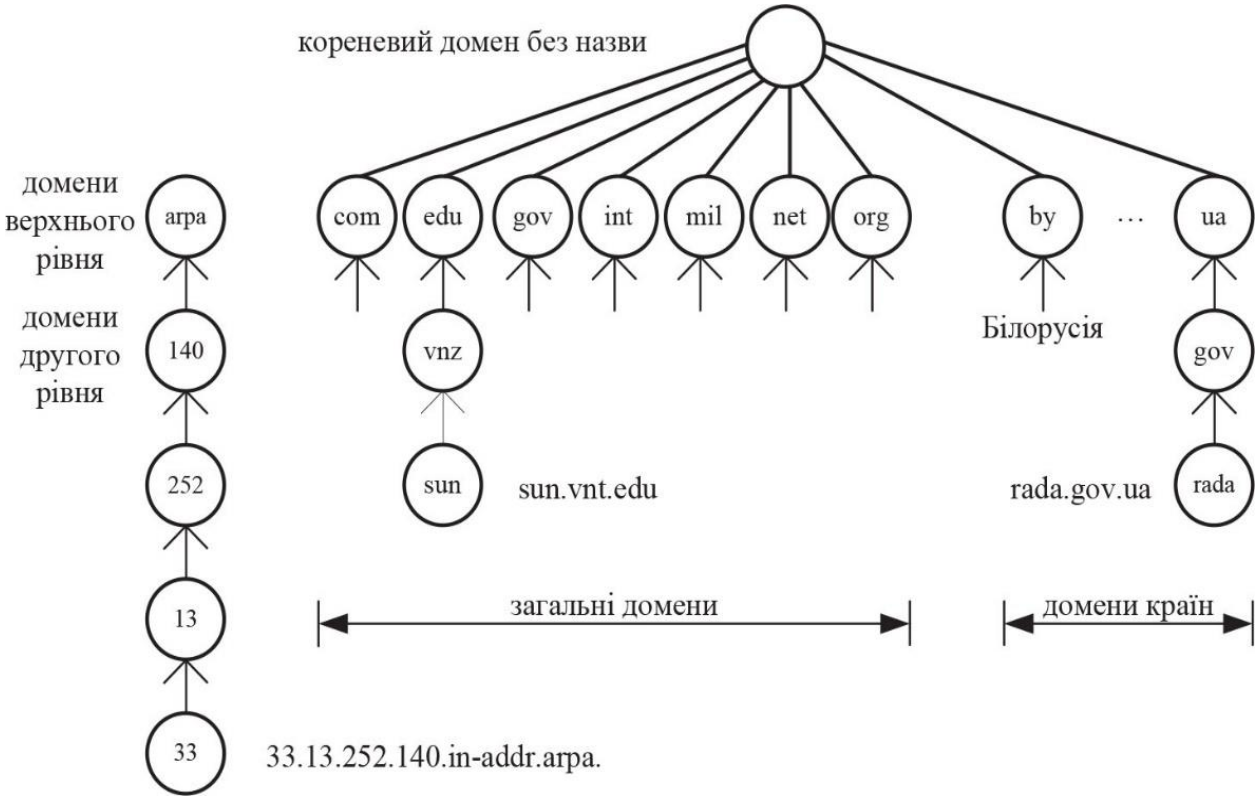


Рисунок 1.1 – Ієрархічна організація DNS

Абсолютним доменним іменем (absolute domain name) називається ім’я домену, що закінчується крапкою, або повним іменем домену (FQDN-fully qualified domain name), наприклад «static.rada.gov.na.»:

1) arpa – це спеціальний домен, що використовується для зіставлення адрес та імен;

2) сім трисимвольних доменів називаються загальними або (general) або організаційними доминами (organizational). Їх класифікацію наведено у таблиці 1.1. Імена .edu. gov.mil зарезервовані за організаціями США, інші можуть використовуватися за межами США. У 2000 р. було сім нових доменів верхнього рівня;

3) всі двосимвольні домени відповідають кодам країн. Їх перелік можна знайти у ISO 3166. Вони називаються доменами країн (county) або географічними (geographical) доменами.

Таблиця 1.1 – Загальні домени

Домен	Опис	Домен (уведено з 2000 р.)	Опис
.com	Комерційні організації	.qero	Галузі, що пов’язані з повітряним транспортом
.edu	Освітні організації	.bis	Організації, що пов’язано з бізнесом
.gov	Урядові організації США	.coop	Некомерційні організації
.int	Міжнародні організації	.into	Для необмеженого використання
.net	Мережі	.museum	Музеї
.org	Інші організації	Name	Окремі особи
.mil	Військові організації США	.pro	Бухгалтери, юристи, лікарі

Передавання відповідальності всередині DNS – це важлива характеристика DNS. Дерево цілком не обслуговує жодна організація. Замість цього одна організація NIC обслуговує лише частину дерева (домени верхнього рівня), а відповідальність за окремі зони передається іншим організаціям. Зона – це окрема керована частина дерева DNS. Відповідальна за керування зоною організація виконує налаштування серверів DNS (name servers) для цієї зони. Коли в зоні з’являється новий вузол (відповідно, нове ім’я), адміністратор зони поміщає ім’я і IP-адресу вузла в базу даних сервера DNS.

Одну або кілька зон може обслуговувати сервер DNS. Саме основний сервер DNS (primary name server) створюється для цієї зони, а також один або кілька вторинних серверів (secondany name server). Щоб система DNS не вийшла з ладу при відмові одного із серверів, первинний і вторинний сервери мають бути незалежними і надлишкованими.

При умові, що сервер DNS не має необхідної інформації, він встановлює контакт з іншим DNS-сервером. Кожен сервер має знати, як встановити контакт з кореневими серверами DNS (root name servers). У свою чергу, кореневі сервери знають імена та IP-адреси кожного офіційного сервера DNS для всіх доменів другого рівня.

Кешування – це фундаментальна характеристика DNS. У момент, коли DNS-сервер отримує інформацію про відповідність імені та адреси, він кешує цю інформацію таким чином, щоб у випадку наступного запиту могла бути використана інформація з кешу. При цьому додатковим запит до інших серверів не виконується.

1.1.2 Формат повідомлення DNS

Основна функція протоколу DNS – керування взаємодією між DNS-клієнтом та DNS-сервером. DNS-клієнт надсилає запит, а DNS-сервер повертає відповідь, що містить необхідну клієнту інформацію. Локальний DNS-сервер надсилає відповіді клієнтам та надсилає запити іншим серверам. Кореневі сервери лише надають відповіді. Наприклад, програма хоче встановити IP-адресу для www.test.site.com. Клієнт зв'язується з локальним DNS-сервером, який звертається до кореневого DNS-сервера, щоб дізнатися IP-адресу DNS-сервера .com.

Далі локальний DNS-сервер відправляє запит DNS-серверу .com., щоб дізнатися IP-адресу DNS-сервера site.com. Після цього локальний DNS-сервер надсилає запит DNS-серверу зони site.com. Якщо зона має підзони, то може бути виданий додатковий запит домену test.site.com. який надасть у відповідь IP-адресу для www.test.site.com.

Всі DNS – запити поділяються на рекурсивні або ітеративні. Рекурсивний запит вимагає, щоб DNS-сервер, який приймає запит, сам виконував перетворення. Наприклад, перетворювач видає рекурсивний запит локальному серверу імен на перетворення доменного імені у IP-адресу. На рисунку 1.2 показано, що на етапі 1 перетворювач активізується через

системний виклик. Далі перетворювач надсилає DNS-запит локальному серверу (етап 2) і чекає відповіді (етап 9). Локальний DNS-сервер здійснює дії з опрацювання запиту перетворювача**.** Ітеративний запит вимагає, щоб даний сервер у відповіді клієнту надав IP-адресу наступного в ієрархії DNS-сервера. Кореневі сервери обслуговують лише ітеративні запити. Локальний DNS-сервер надсилає запит кореневому DNS - серверу (етап 3), щоб дізнатися ім’я та IP-адресу DNS-сервера для зони на наступному рівні ієрархії (етап 4). Це допомагає розвантажити кореневі сервери. Локальний DNS-сервер може відправити запит наступному серверу в ієрархії (етапи 5, 6, 7, 8). Нарешті локальний сервер відповідає перетворювачу (етап 9), а перетворювач надає IP-адресу застосуванню (етап 10).

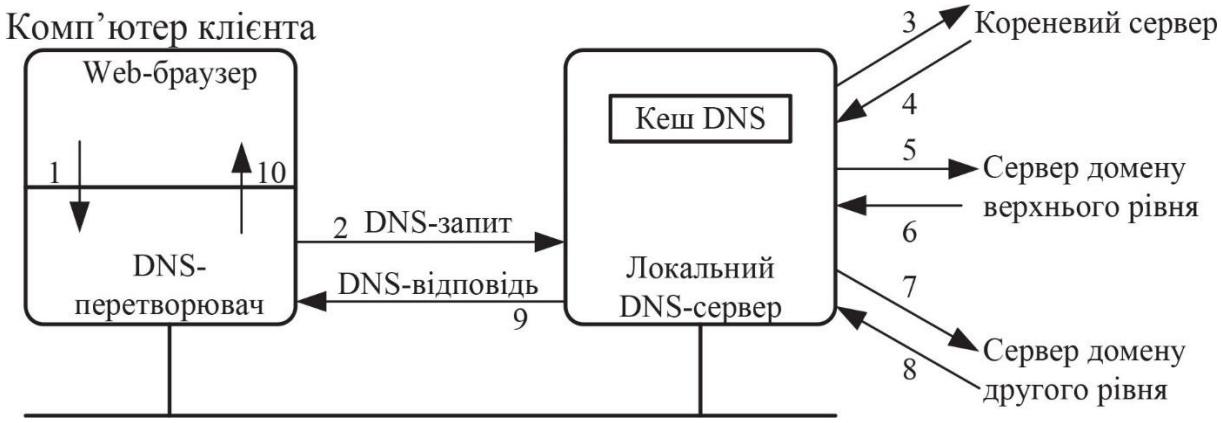


Рисунок 1.2 – Опрацювання DNS-запиту

Однаковий формат використовується для DNS - запитів і DNS - відповідей. На рисунку 1.3 показано загальний формат DNS-повідомлення.

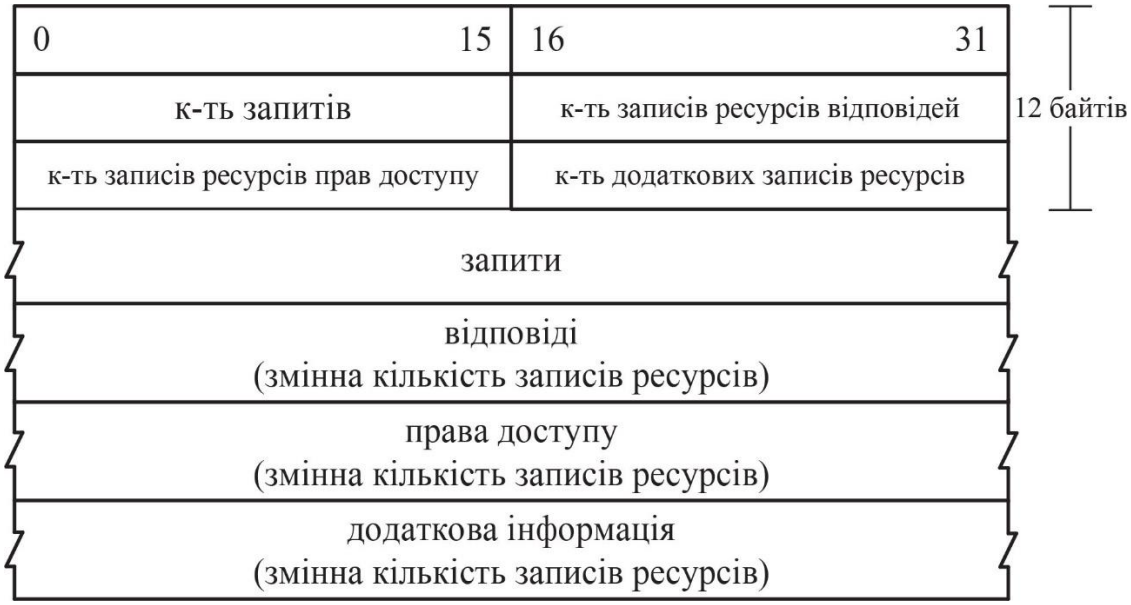


Рисунок 1.3 – Формат DNS-повідомлення

Фіксований 12-байтовий заголовок містить повідомлення, за яким йде чотири поля змінної довжини значення полів:

1) поле ідентифікації (identification) встановлюється клієнтом і повертається сервером. Це поле дозволяє клієнту визначити, на який запит прийшла відповідь:

2) 16-бітове поле прапорців (flags) поділено на кілька частин, як показано на рисунку 1.4.

1	4	1	1	1	1	3	4
QR	opcode	AA	TC	RD	RA	zero	rcode

Рисунок 1.4 – Поле прапорців заголовка DNS-повідомлення

Тут:

– QR (тип повідомлення), 1-бітове поле; 0 означає запит, 1 – відповідь;

– OP code (код операції), 4-бітове поле. Як правило, містить значення 0 (стандартний запит). Інші значення – це 1 (інверсний запит) і запит статусу сервера;

– AA– 1-бітовий прапорець, що означає «авторитетна відповідь» (authonitavsve answer). Сервер DNS має повноваження для цього домену у розділі запитів;

– TC- 1-бітове поле, що означає «обрізано» (truncated). Для DNS це означає, що повний розмір відповіді перевищує 512 байтів, але було повернуто лише перші 512 байтів відповіді;

– RD – 1-бітове поле, що означає «необхідна рекурсія» (recursion desired). Біт може бути встановлений у запиті, а потім повернутий у відповіді. Цей прапорець вимагає від DNS-сервера опрацювати даний запит як рекурсивний (recursive duery), тобто сервер має сам визначити необхідну IP-адресу, а не повертати адресу іншого DNS-сервера. Якщо даний біт не встановлено і DNS-сервера, що отримав запит, не має авторитетної відповіді, він повертає у відповіді список інших DNS-серверів, до яких необхідно звернутися, щоб отримати відповідь. Це називається запитом, що повторюється (iterative duery);

– RA – 1-бітове поле, що означає «рекурсія можлива» (recursion available). Цей біт встановлюється в 1 у відповіді, якщо сервер підтримує рекурсію. Більшість серверів DNS підтримує рекурсію, за винятком кількох кореневих серверів, які є надто завантаженими;

– zero –3-бітове поле, що має дорівнювати 0;

– rcode – це 4-бітове поле коду відповіді. Звичайні значення: 0 (нема помилок) і 3 (помилки імені). Помилка імені повертається тільки від авторитетного DNS-сервера і означає, що імені домену, яке вказано у запиті, не існує;

3) Наступні чотири 16-бітових поля вказують на кількість пунктів у чотирьох полях змінної довжини, що закінчують повідомлення. У запиті кількість запитань зазвичай дорівнює 1, а інші три лічильника дорівнюють 0. У запиті, що повертається, кількість відповідей дорівнює, як мінімум 1, а інші можуть бути як нульовими так і ненульовими. Формат кожного DNS-

запиту у полі запитання показано на рисунку 1.5 (зазвичай є лише одне запитання).

Рисунок 1.5 – Формат поля DNS-повідомлення

Тут:

– Ім’я запиту – це ім’я, що шукається. Воно виглядає як послідовність з однієї або кількох міток. Кожна мітка починається з 1-байтового лічильника, який містить кількість байтів, що йдуть за ним. Ім’я закінчується байтом, що дорівнює 0. Він є міткою з нульовою довжиною, а також міткою кореня. Кожний лічильник байтів має бути в діапазоні від 0 до 63, оскільки довжина мітки обмежується 63 байтами. Це поле може закінчуватися на обмежувачі, що не дорівнює 32 байтам. Заповнення не використовується. На рисунку 1.6 показано, як зберігається ім’я домену.

4	t	e	s	t	6	d	o	m	a	i	n	3	e	d	u	2	u	a	0

Рисунок 1.6 – Зберігання імені домену test. domanin. edu. га у частині ім’я поля запитання DNS-запиту

– у кожного запитання є тип запиту (query type), а також відповідь має тип (type). Існує біля 20 різних значень, частина з яких уже застаріла. У таблиці 1.2 наведено деякі з цих значень. Тип запиту– це надмножина (множина, підмножиною якого є дана множина) типів: два з наведених значень можуть використовуватися лише у запитаннях. Найбільш поширений тип запису – тип А, що означає, що необхідна IP-адреса для вказаного імені (query name). PTR-запит вимагає імені, що відповідає IP-адресі.

Таблиця 1.2 – Значення типу запиту поля запитання DNS-повідомлення

Ім’я	Цифрове значення	Опис	Тип (type)	Тип запиту (query type)
A	1	IP-адреса	Так	Так

NS	2	Сервер- DNS	Так	Так
CNAME	5	Канонічне ім’я	Так	Так
PTR	12	Запис вказівника	Так	Так
NINFO	13	Інформація про вузол	Так	Так
MX	15	Запис про обмін поштою	Так	Так
AXFR	252	Запит про передавання зони	Ні	Так
ANY	255	Запит всіх записів	Ні	Так

– клас запиту (query class) звичайно дорівнює 1, що вказує на адреси Internet;

4) останні три поля DNS-повідомлення– це відповідні (answers), права (authority) і додаткова інформація (additional information). Загальний формат називається записом ресурсу (RR-resource record).

На рисунку 1.7 показано загальний формат запису ресурсу.

<table>
<tr><td>0 15</td><td>16 31</td></tr>
<tr><td colspan="2">ім’я домену</td></tr>
<tr><td>тип</td><td>клас</td></tr>
<tr><td colspan="2">час життя</td></tr>
<tr><td>довжина даних ресурсів</td><td rowspan="2"></td></tr>
<tr><td>дані курсу</td></tr>
</table>

Рисунок 1.7 – Загальний формат запису ресурсу

Проаналізуємо відповідні поля на рисунку 1.7:

– тип (type) вказує на один із типів кодів RR. Це те саме, що і значення типу запиту. Для даних мережі Internet поле клас (class) звичайно встановлено в 1;

– ім’я домену (domain name) – це ім’я, якому відповідають наступні дані ресурсу. Формат імені такий же, як показано на рисунку 6.7 для поля запитання;

– поле час життя (TTL-time-to-live) – це кількість секунд, протягом яких RR може бути кешованим клієнтом. Звичайно RR дорівнює 2 дням;

– довжина запису ресурсу (resource data length) вказує на кількість даних ресурсу (resource data). Для типу, що дорівнює 1 (запис А), дані ресурсу –це 4-байтова IP-адреса.

1.2 Принципи функціонування протоколу DNS

Як засобами локального вузла, так і засобами централізованої служби може встановлюватися відповідність між доменними іменами (FQDN) і IP – адресами [5]. Найпростішим рішенням є створення вручну текстового файлу з ім'ям hosts. Цей файл складається з деякої кількості рядків, кожна з яких містить запис типу «IP–адреса – доменне ім'я», наприклад, 77.47.129.30 – www.kpi.ua. Файл необхідно розмістити у відповідному системному каталогу (наприклад, у випадку Windows – систем це каталог \WINDOWS\system32\drivers\etc).

Загалом, DNS використовується для вирішення проблеми збігу доменних імен та IP-адрес). Служба DNS використовує у своїй роботі схему взаємодії клієнт-сервер. В рамках цієї взаємодії визначаються DNS-сервери та DNS-клієнти. DNS-сервери підтримують розподілену базу даних записів, а DNS-клієнти звертаються до серверів із запитами визначення IP-адреси для конкретного доменного імені вузла чи ресурсу. Таким чином, DNS надає механізми як для іменування вузлів, так і для пошуку IP-адрес вузлів на ім'я.

Майже такого ж формату служба DNS використовує текстові файли, як і файл hosts, і ці файли адміністратор також може корегувати вручну. Проте, служба DNS спирається на ієрархію доменів, і кожен сервер служби DNS зберігає тільки частину імен мережі, а не всі імена, як це відбувається при використанні файлів hosts. При зростанні кількості вузлів в мережі проблема масштабування вирішується створенням нових доменів і піддоменів імен і додаванням в службу DNS нових серверів.

Свій DNS–сервер створюється для кожного домена імен. Цей сервер може зберігати записи «доменне ім'я – IP–адреса» для всього домена, включаючи всі його піддомени. Частіше сервер домена зберігає тільки імена, які закінчуються на наступному нижче рівні ієрархії в порівнянні з ім'ям домена. Саме при такій організації служби DNS навантаження розподіляється більш–менш рівномірно між всіма DNS–серверами мережі. Наприклад, в

першому випадку DNS – сервер домена kpi.ua зберігатиме відображення для всіх імен, що закінчуються на kpi.ua. В другому випадку цей сервер зберігає відображення тільки імен типу fel.kpi.ua, fiot.kpi.ua, а решта всіх записів нижчих рівнів ієрархії повинна зберігатися на DNS–серверах піддоменів fel і fiot.

Необхідно правильно сконфігурувати DNS–сервери, зони і зробити необхідні записи для нормальної роботи DNS [5].

DNS–сервер – це комп'ютер з відповідними програмними додатками, наприклад, служба DNS–сервера в Windows, або служба BIND в UNIX–системах. DNS–сервери підтримують базу даних DNS з інформацією про частини структури доменного дерева DNS і обробляють запити на дозвіл імен, що надходять від DNS–клієнтів. У відповідь на запит клієнта DNS–сервер надає запитувану інформацію, дає посилання на інший сервер, який може відповісти на запит, або повідомляє, що інформація недоступна або не існує. DNS–сервери поділяються на основні (повноважні) та резервні (додаткові).

Для кожної зони може існувати тільки один основний сервер (на якому можна вносити зміни в зонну інформацію) і будь–яка кількість резервних серверів (які отримують інформацію про зону з основного серверу). Встановлення резервних DNS серверів дозволяє вирішити дві задачі: збільшити надійність роботи системи DNS, розподілити запити клієнтів по різним серверам, збільшуючи таким чином продуктивність роботи системи DNS.

Зона DNS (DNS zone) – це єдина частина простору імен та IP – адрес, що обслуговується повноважним сервером [5]. Сервер може обслуговувати і кілька зон, а зона може містити один або декілька Інтернет – доменів. Наприклад, один сервер може бути повноважним для зон kpi.ua і kpi.edu, кожна з яких містить декілька доменів. Суміжні домени, наприклад, kpi.ua, fel.kpi.ua і keoa.fel.kpi.ua можна перетворити в окремі зони, застосувавши

делегування, при якому відповідальність за піддомен всередині простору імен DNS присвоюється окремому об'єкту (відповідному DNS – серверу).

Файли зон (zone files) містять записи ресурсів зон, в яких сервер є повноважним. У багатьох реалізаціях DNS–сервера дані зон зберігаються в текстових файлах; DNS–сервери на контролерах доменів під керуванням Windows 2000 або Windows Server 2003 можуть також зберігати зонну інформацію в Active Directory.

Існують два види зон: прямого і зворотного перегляду [5]. У перших виконується зіставлення FQDN–імен з IP–адресами, у других навпаки, IP–адреси зіставляються з повними доменними іменами. Таким чином, зони прямого перегляду обслуговують запити по встановленню відповідності між FQDN–іменами і IP–адресами, а зони зворотного перегляду – між IP–адресами і FQDN–іменами.

Якщо ім’я прямої зони співпадає з іменем домену, то ім’я зворотної зони формується з мережної частини IP–адреси, записаної в зворотному порядку, до якої додається стандартний префікс – in–addr.arpa. Наприклад, кафедрі КЕОА виділено блок IP–адрес 10.12.80.0/24. Ім’я зворотної зони – 80.12.10. in–addr.arpa. Існують поняття основної зони, додаткової зони і зони – заглушки (рис. 1.8).

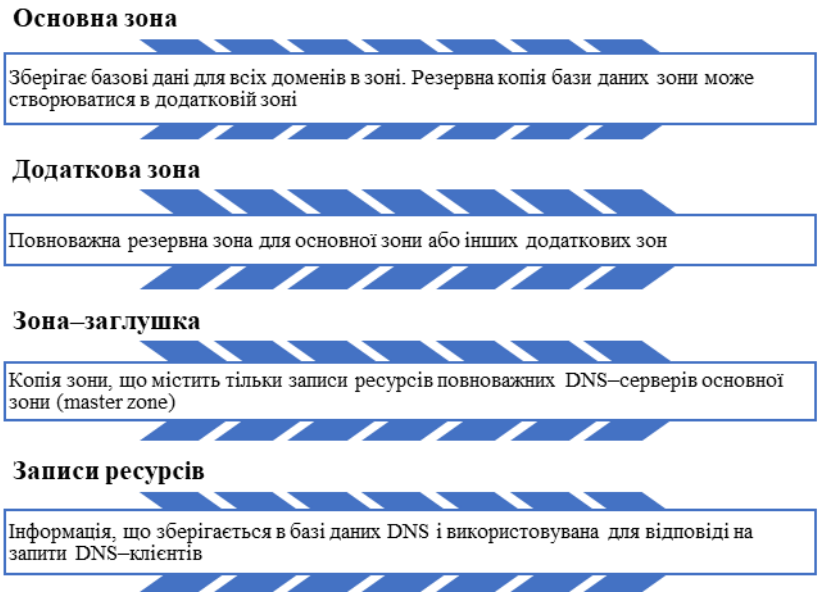


Рисунок 1.8 – Зони DNS

Кожен DNS–сервер містить записи ресурсів, необхідні йому для відповіді на запити, що відносяться до його частини простору імен DNS. Записи ресурсів розрізняються за типами: наприклад, адресний запис (A), канонічне ім'я (CNAME), сервер імен (NS), поштовий обмінник (MX).

Найбільш важливі типи DNS–записів (рис. 1.9).

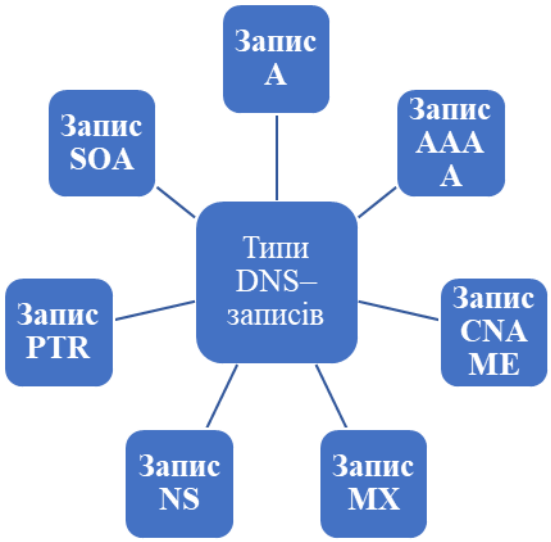


Рисунок 1.9 – Типи DNS–записів

Запис A (address record) або запис адреси – зв'язує ім'я вузла з IP – адресою. Наприклад, запит A–запису на ім'я www.kpi.ua поверне його IP–адресу – 10.7.10.22.

Запис AAAA (IPv6 address record) зв'язує ім'я вузла з адресою протоколу IPv6.

Запис CNAME (canonical name record) або канонічний запис імені (псевдонім) використовується для перенаправлення на інше ім'я. Дозволяє одній IP– адресі поставити у відповідність декілька імен.

Запис MX (mail exchange) або поштовий обмінник – вказує сервер (и) обміну поштою для даного домену.

Запис NS (name server) – вказує на DNS–сервер для даного домену.

Запис PTR (pointer) або запис вказівника – зв'язує IP – адресу вузла з його канонічним ім'ям. Використовується у зворотній зоні і є аналогом A–запису у прямій зоні. Запит PTR запису в зворотній зоні in–addr.arpa на IP –

адресу вузла поверне ім'я (FQDN) даного вузла. З метою зменшення обсягу небажаної кореспонденції (спаму) багато серверів–одержувачів електронної пошти можуть перевіряти наявність PTR запису для вузла, з якого відбувається відправлення. У цьому випадку PTR запис для IP – адреси повинен відповідати імені відправляючого поштового сервера, яким він представляється в процесі SMTP сесії.

Запис SOA (Start of Authority) або початковий запис зони – вказує на основний DNS – сервер зони, на якому зберігається еталонна інформація про зону, містить контактну інформацію особи, відповідальну за дану зону, задає параметри (серійний номер та часові параметри), необхідні для оновлення інформації про зону резервними DNS–серверами.

SRV–запис (server selection) вказує на сервери для деяких сервісів; використовується, зокрема, для Jabber і Active Directory.

Запис TXT (Text) – текстовий запис, який використовується для внесення коментарів, поміток і тощо.

Програма named, яка реалізує BIND, як і довільний сервіс прикладного рівня використовує транспорт TCP та UDP. Якщо ми не можемо звернутись до певного комп’ютера, а раніше могли, то перш за все треба перевірити доступність комп’ютера за його IP адресою. Якщо цього не можна зробити, то треба шукати помилки або збої в роботі сервісу доменних імен. Клієнтська частина – це процедура дозволу імен resolver, а сервер – програма named. Resolver – це набір процедур з системи бібліотеки libc.a які дозволяють прикладній програмі, яка відредагована з цими процедурами отримати за доменним іменем IP адресу комп’ютера, або за IP адресою доменне ім’я. Ці процедури звертаються до системної компоненти resolver, яка веде діалог із сервером доменних імен і таким чином обслуговує запити прикладних програм користувача. Загальну схему взаємодії різних компонент BIND можна представити на рисунку 1.10.

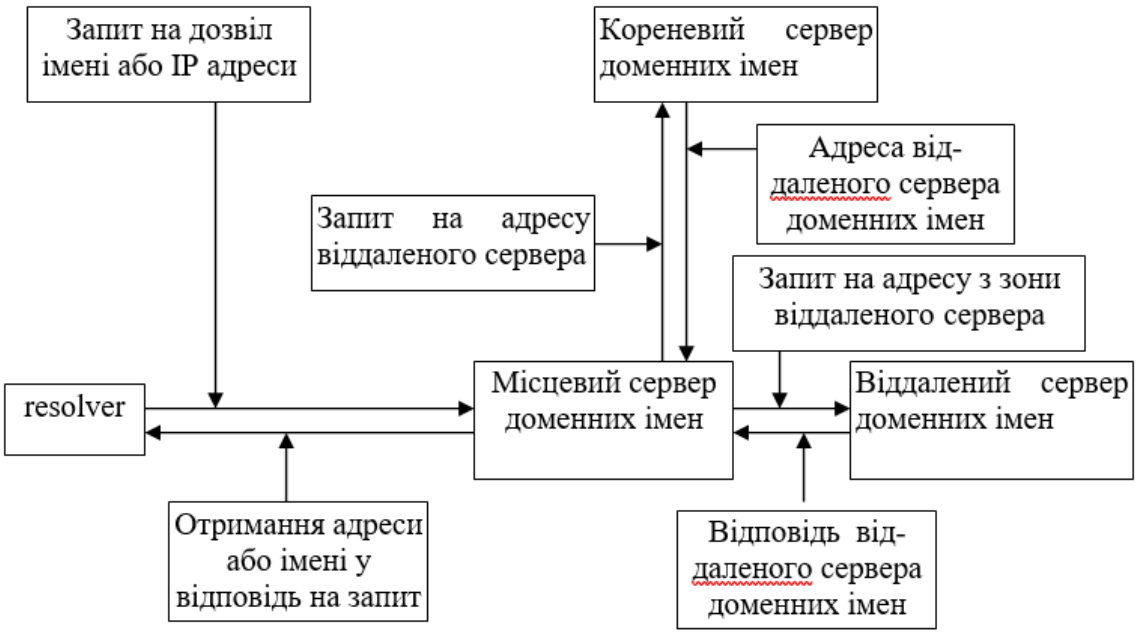


Рисунок 1.10 – Взаємодія компонент BIND

Спеціальною базою даних керується named, яка складається з декількох файлів і містить відповідності між адресами і іменами, а також адреси інших серверів BIND, до яких даних сервер може звертатись в процесі пошуку імені або адреси. Опираючись риснок 1.10 розглянемо два способи дозволу запиту на отримання IP адреси за доменним іменем.

1-й випадок. Запит на отримання IP адреси в рамках зони відповідальності даного місцевого сервера імен.

1) Прикладна програма через resolver запитує IP адресу за доменним іменем у місцевого сервера.

2) Місцевий сервер повідомляє прикладній програмі IP адресу запитаного імені.

2-й випадок. Запит прикладної програми до сервера доменних імен на отримання IP адреси за доменним іменем із домена, який знаходиться в зоні відповідальності віддаленого сервера доменних імен, тобто сервера відмінного від того, домену якого належить комп’ютер який виконує запит.

В загальному випадку схема виглядає наступним чином:

1) Прикладна програма звертається до місцевого сервера доменних імен за IP адресою, повідомляючи йому доменне ім'я.

2) Сервер визначає, що адреса не входить в даний домен і звертається за адресою сервера домена , який запитує до кореневого сервера доменних імен.

3) Кореневий сервер доменних імен повідомляє місцевому серверу доменних імен адресу сервера того домена, який вимагається.

4) Місцевий сервер доменних імен запитує віддалений сервер на предмет дозволу запиту свого клієнта – прикладної програми.

5) Віддалений сервер повідомляє IP адресу місцевому серверу.

Місцевий сервер повідомляє IP адресу прикладній програмі.

1.3 Проблеми функціонування DNS-служби

Згадаймо DNS-алгоритм віддаленого пошуку IP-адреси на ім'я в Мережі:

– хост посилає на IP-адресу DNS-сервера свого домену (він задається при налаштуванні протоколу IP у мережній ОС) DNS-запит, в якому вказує ім'я сервера, IP-адресу якого необхідно знайти;

– DNS-сервер, отримавши запит, переглядає свою базу імен на наявність у ній імені, що міститься в запиті. У випадку, якщо ім'я знайдено, а отже, знайдено і відповідну йому IP-адресу, на хост, що запитав DNS-сервер відправляє DNS-відповідь, в якому записана шукана IP-адреса. Якщо вказане в запиті ім'я DNS-сервер не виявив у своїй базі імен, то DNS-запит надсилається DNS-сервером на один із кореневих DNS-серверів, адреси яких містяться у файлі налаштувань DNS-сервера root. cache, та описана в цьому пункті процедура повторюється, доки ім'я не буде знайдено.

Аналізуючи з погляду безпеки вразливість цієї схеми віддаленого пошуку з допомогою протоколу DNS, можна дійти невтішного висновку про можливість некоректного функціонування DNS-сервісу, саме можна

виділити основні причини неправильного функціонування:

– віддалена атака – «Помилковий об'єкт РВС» (розподіленої обчислювальної системи), тобто. впровадження проміжного хоста, через який йтиме потік інформації між атакованим об'єктом і сервером або підміна (виправлення) інформації про зону;

– міжсегментна віддалена атака – атака на DNS шляхом фальсифікації відповіді DNS – сервера;

– хибні дії адміністратора DNS-сервера, тобто. неправильна вказівка відповідності між IP-адресою хоста та його ім'ям.

Віддалені атаки на DNS-сервер. Практичні пошуки та критичний аналіз безпеки служби DNS дозволяють припустити, що існують, як мінімум, два можливі варіанти віддаленої атаки з використанням помилкового об'єкта на цю службу:

«Шторм» помилкових відповідей DNS. Перший варіант проведення віддаленої атаки, спрямованої на службу DNS, заснований на різновиді типової віддаленої атаки «Помилковий об'єкт РВС» [2]. У цьому випадку атакуючий здійснює постійну передачу на хост, що атакується, заздалегідь підготовленого помилкового DNS-відповіді від імені справжнього DNS-сервера без прийому DNS-запиту (рис 1.11).

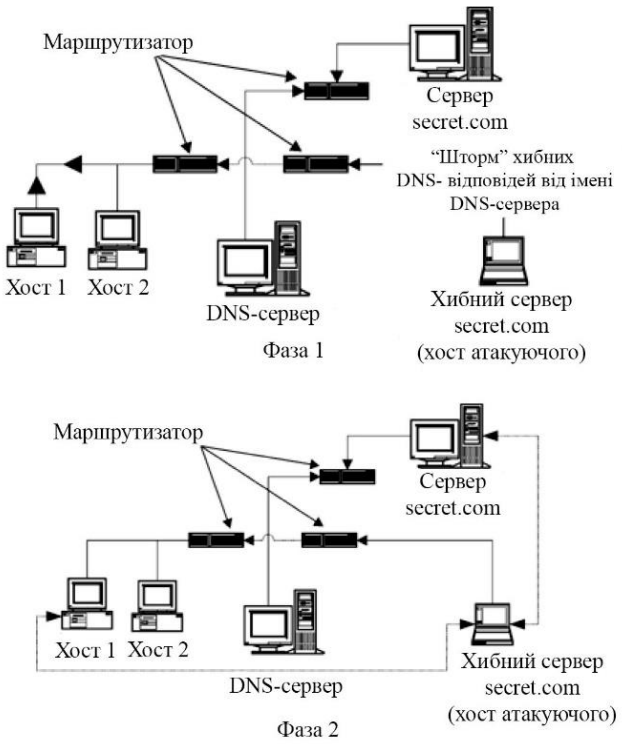


Рисунок 1.11 – «Шторм» хибних відповідей

Іншими словами, атакуючий створює в мережі Internet спрямований «шторм» помилкових DNS-відповідей. Це можливо, оскільки зазвичай передачі DNS-запиту використовується протокол UDP, у якому відсутні засоби ідентифікації пакетів. Критерії, що пред'являються мережевий ОС хоста до отриманої від DNS-сервера відповіді, – це, по-перше, збіг IP-адреси відправника відповіді з IP-адресою DNS-сервера; по-друге, необхідно, щоб у DNS-відповіді було вказано те саме ім'я, що й у DNS-запиті; по-третє, DNS-відповідь має бути спрямований на той же UDP-порт, з якого був надісланий DNS-запит, і, по-четверте, у DNS-відповіді поле ідентифікатор запиту в заголовку DNS (ID) має містити те саме значення, що й у переданому DNS-запиті.

Перехоплення запиту DNS. З розглянутої раніше схеми віддаленого DNS-пошуку випливає, що в тому випадку, якщо вказане в запиті ім'я DNS-сервер не виявив у своїй базі імен, запит надсилається сервером на один з кореневих DNS-серверів, адреси яких містяться у файлі налаштувань сервера root. Тобто у тому випадку, якщо DNS-сервер не має відомостей про запит

хості, він пересилає запит далі – це означає, що тепер сам DNS-сервер є ініціатором віддаленого DNS-пошуку. Тому ніщо не заважає атакуючій, діючи описаними вище методами, перенести свій удар безпосередньо на DNS-сервер [2]. Інакше кажучи, як мета атаки тепер виступатиме не хост, а DNS-сервер і помилкові DNS-відповіді будуть направлятися атакуючим від імені кореневого DNS-сервера на атакований DNS-сервер. При цьому важливо враховувати таку особливість роботи сервера DNS. Для прискорення роботи кожен DNS-сервер кешує в області пам'яті свою таблицю відповідності імен та IP-адрес хостів. У тому числі в кеш заноситься інформація, що динамічно змінюється, про імена та IP-адреси хостів, знайдених у процесі функціонування DNS-сервера. Тобто. якщо DNS-сервер, отримавши запит, не знаходить у себе в кеш-таблиці відповідного запису він пересилає відповідь на наступний сервер і, отримавши відповідь, заносить знайдені відомості в кеш-таблицю. Таким чином, при отриманні наступного запиту DNS-сервер вже не потрібно вести віддалений пошук, так як необхідна інформація вже знаходиться в пам'яті.

З аналізу описаної схеми віддаленого DNS-пошуку стає очевидно, що в тому випадку якщо у відповідь на запит від DNS-сервера атакуючий направить помилкову DNS-відповідь (або у разі «шторму» помилкових відповідей буде вести їх постійну передачу), то в кеш- таблиці сервера з'явиться відповідний запис з неправдивими відомостями, і надалі всі хости, що звернулися до цього DNS-сервера, будуть дезінформовані, і при зверненні до хоста, маршрут до якого атакуючий вирішив змінити, зв'язок з ним здійснюватиметься через хост атакуючого за схемою «об'єкт РВС». І з часом ця хибна інформація, що потрапила в кеш DNS-сервера, поширюватиметься на сусідні DNS-сервери найвищих рівнів.

Віддалена міжсегментна атака на DNS-сервер. Значно загальнішим випадком є міжсегментна атака, що не вимагає для своєї реалізації таких жорстких умов, коли атакуючий і цільовий DNS-сервер поділяють загальне фізичне середовище передачі.

Міжсегментна атака на DNS-сервер має такий вигляд. Припустимо, що метою атаки є «підміна» IP-адреси web-сервера www.coolsite.com на IP-адресу сервера www.badsite.com для користувачів деякої підмережі, яку обслуговує DNS-сервер ns.victim.com. У першій фазі атаки ns.victim.com провокується на пошук інформації про IP-адресу www.coolsite.com шляхом надсилання йому відповідного рекурсивного запиту. У другій фазі атакуючий надсилає серверу ns.victim.com складну відповідь від імені ns.coolsite.com, яка є відповідальною за домен coolsite.com. У хибній відповіді замість реальної IP-адреси www.coolsite.com вказується IP-адреса www.badsite.com. Сервер ns.victim.com кешує отриману інформацію, в результаті чого протягом певного проміжку часу (величина цього проміжку вказується в полі TTL помилкової відповіді і може довільно вибиратися атакуючим) користувачі, що нічого не підозрюють, замість сервера www.coolsite.com потрапляють на www.badsite .com (рис. 1.12).

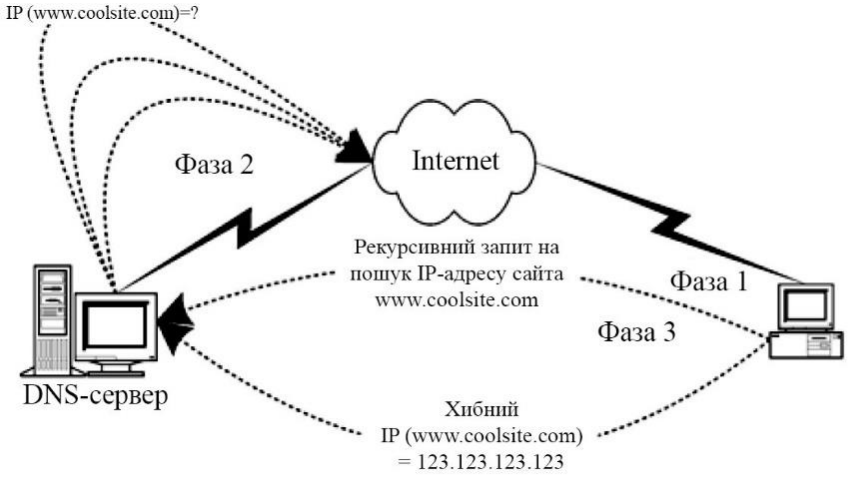


Рисунок 1.12 – Міжсегментна віддалена атака

Для того, щоб помилкова відповідь була сприйнята сервером ns.victim.com як істинна, достатньо виконання чотирьох умов:

– IP-адреса відправника відповіді повинна відповідати IP-адресі запитуваного сервера (в даному випадку ns.coolsite.com);

– UDP-порт, на який надсилається відповідь, повинен збігатися з портом, з якого було надіслано запит;

– ідентифікатор відповіді має збігатися з ідентифікатором запиту;

– відповідь повинна містити запитувану інформацію (у цьому випадку IP-адреса web-сервера www.coolsite.com).

Очевидно, що виконання першої та четвертої умов не представляє для атакуючого особливих труднощів. З другою та третьою умовами ситуація набагато складніша, оскільки у випадку міжсегментної атаки у атакуючого немає можливості перехопити вихідний запит і «підглянути» необхідні параметри.

Помилкові дії адміністратора DNS-сервера. Помилкові дії адміністратора сервера DNS, тобто. неправильне вказівку відповідності IP-адреси хоста та його імені можуть призвести до поширення помилки на інші DNS-сервера. Отже, при зверненні до DNS-сервера він видаватиме неправильну IP-адресу шуканого хоста.

Як видно з розділу – у мережі існує достатньо проблем, пов'язаних з коректністю функціонування DNS-служби, і це досить серйозні проблеми, які можуть ускладнити роботу користувачів і мережевих адміністраторів. У наступному розділі будуть розглянуті деякі сучасні рішення та поради для уникнення цих проблем.

Деякі рішення проблем функціонування служби DNS. Оптимальним з точки зору безпеки рішенням взагалі відмовитися від використання служби DNS в сегменті, що захищається. Звичайно, зовсім відмовитись від використання імен при зверненні до хостів для користувачів буде дуже не зручно. Тому можна запропонувати таке компромісне рішення: використовувати імена, але відмовитися від механізму віддаленого DNS-пошуку, який використовувався до появи служби DNS із виділеними DNS-серверами. Тоді на кожній машині в мережі існував hosts файл, в якому знаходилася інформація про відповідні імена та IP-адреси всіх хостів у мережі. Очевидно, що на сьогоднішній день адміністратору можна внести в подібний файл інформацію про лише найчастіше відвідувані користувачами даного сегмента сервери мережі.

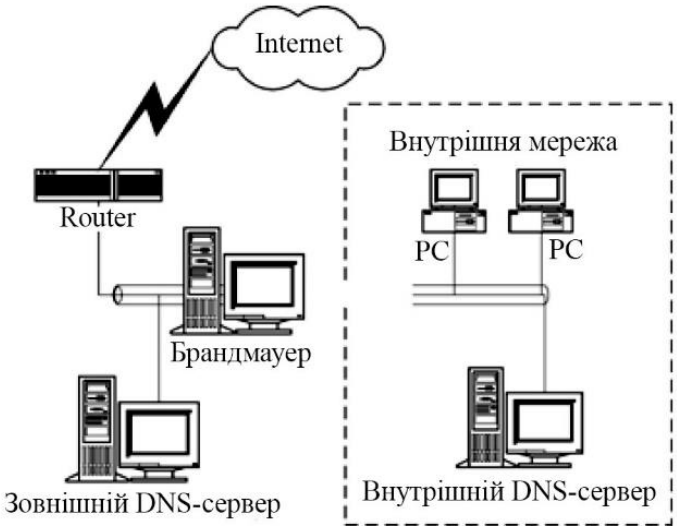


Рисунок 1.13 – Внутрішній сервер DNS обслуговує корпоративну мережу і не видно ззовні. Зовнішній DNS-сервер надає лише частину інформації про мережу

Для утруднення здійснення віддаленої атаки можна запропонувати адміністраторам використовувати для служби DNS замість протоколу UDP, який встановлюється за замовчуванням, протокол TCP (хоча з документації не очевидно, як його змінити). Це суттєво утруднить для атакуючого передачу на хост помилкового DNS-відповіді без прийому DNS-запиту.

Також можна запропонувати використовувати комплект додатків BIND (Berkley Internet Name Daemon) – берклівський демон імен Internet. Починаючи з версії 4.9.3, у специфікацію BIND було внесено кілька директив і типів записів DNS, які покликані дещо покращити захист серверів імен. Директива xfrnets файлу початкового завантаження (/etc/named.boot) дозволяє вказати список IP-адрес мереж і серверів імен, на які даний сервер має право пересилати інформацію по зоні (операція зонного пересилання). Друга важлива новація – введення особливого типу запису ресурсів TXT під назвою SECURE_ZONE. Такий запис управляє списком машин і мереж (за IP-адресами), яким можна запитувати даний сервер імен. Але незважаючи на ці нововведення для відображення атак із заміною DNS потрібно вжити ще низку заходів. Серед них найбільш поширеною є установка двох серверів

DNS: зовнішнього та внутрішнього (див. рис. 1.8).

Внутрішній DNS-сервер призначений виключно для обслуговування внутрішніх клієнтів мережі. На ньому зберігається вся інформація про хості корпоративної мережі. Завдяки використанню записів типу SECURE_ZONE цей сервер можуть вимагати лише внутрішні хости. Більш того, на брандмауері встановлюється фільтр, який не пропускає IP-пакети, що направляються в корпоративну мережу та призначені для порту 53 протоколів UDP та TCP внутрішнього сервера DNS. Тобто. зовні такий сервер DNS робиться невидимим. Однак він сам може звертатися за інформацією до серверів DNS мережі Internet.

Остання версія BIND 8.2.2. включає підтримку (RFC 2065) криптографічного цифрового підпису, тобто. це вже не стандартний DNS-протокол, а розширений, в якому в тіло DNS-запиту включатиметься цифровий підпис. Це рішення практично повністю убезпечить роботу із DNS-службою. На жаль, бажаний результат може дати лише широкомасштабне впровадження нових протоколів, яке пов'язане зі значними організаційними труднощами та не може бути проведене за короткий час.

Висновки за розділом

DNS - це система, яка дозволяє користувачам вводити зрозумілі для людини назви сайтів у пошуковий рядок, а браузеру отримувати IP-адресу ресурсу, до якого потрібно звернутися. Система зберігання DNS-даних (зон) розподілена, а ось сервери, на яких зберігаються дані, вибудовані ієрархічно. На верхньому рівні знаходяться кореневі DNS-сервери, нижче - DNS сервери географічних зон, ще на нижчий рівень - локальні DNS-сервери. DNS-запит браузером посилається спочатку на локальні DNS-сервери (resolve), а потім вони шукають потрібну DNS-запис по ієрархії вище. DNS-дані кожного домену називаються DNS-зонами і зберігають DNS-записи різних типів. DNS-зонами можна керувати за допомогою платних та безкоштовних сервісів.

РОЗДІЛ 2
ДОСЛІДЖЕННЯ ПРОЦЕДУР ОБМІНУ ПОВІДОМЛЕНЬ DNS (АНАЛІЗ ПРОТОКОЛІВ)

2.1 Значення мережевих аналізаторів та їх особливості

Аналізатори мереж, також відомі як аналізатори трафіку, протоколів чи пакетів, є програмами чи пристроями, призначеними для моніторингу та аналізу потоку даних у мережі. Ці інструменти можуть обробляти необроблені двійкові дані, перетворюючи їх у зрозумілий формат, що спрощує подальший аналіз мережі.

Законне використання аналізатора мережі включає в себе вирішення проблем, контроль за мережевою безпекою та надання допомоги адміністраторам мережі. Однак існує потенціал для нелегального використання, особливо у випадках, коли хакери намагаються отримати конфіденційну інформацію чи незаконно отримати доступ до мережі без відома IT-адміністратора.

Вбудована функція "фільтра" у мережевих адаптерах Ethernet призначена для ігнорування трафіку, який не призначений для конкретного адаптера. Кадри, які не відповідають призначенню, відхиляються, навіть якщо мережевий адаптер перебуває в "безладному режимі" і отримує всі кадри, навіть якщо MAC-адреса не відповідає йому.

Різні компоненти мережевих аналізаторів, такі як декодери, буфери, фільтри захоплення та обладнання, виконують важливі функції. Процедура "перехоплення" мережі описана нижче:

1. Збір: Перший етап роботи мережевих аналізаторів - це збір даних. Мережеві аналізатори налаштовують мережеві карти інтерфейсу (NIC) в "безладному режимі". Це дозволяє NIC цього комп'ютера фіксувати та

слухати весь трафік у своєму сегменті мережі, зберігаючи необроблені двійкові дані.

2. Перетворення: Другий етап включає процес перетворення, який здійснюється компонентом декодера пакетів. Цей етап перетворює зібрані необроблені двійкові дані в читабельний формат.

3. Аналіз: Останній крок - це аналіз, який включає в себе процес прослуховування (аналіз протоколу). Протоколи, які були зібрані на попередньому етапі та використовуються у мережевому трафіку, можуть бути розглянуті та проаналізовані для отримання інформації. З точки зору протоколів, всі пакети можуть бути інтерпретовані та проаналізовані.

2.2 Загальне використання

Мережеві аналізатори можуть бути використані як для легальних, так і для незаконних завдань. У законному використанні вони можуть приносити безліч переваг мережі. Адміністратори мережі активно використовують ці засоби для оптимізації мережі, виявлення та усунення проблем, а також для захисту від порушень та затримок. Ці інструменти допомагають адміністраторам ефективно вивчати ефективність брандмауерів, списків контролю доступу та функціоналу протоколів.

Водночас вони можуть використовуватися для здійснення незаконних дій, таких як крадіжка конфіденційної інформації чи підслуховування мережі хакерами. Така діяльність, безсумнівно, карається законом. З метою захисту та шифрування мережі використовуються криптографічні протоколи, такі як SSL, SSH, TLS тощо, хоча різні атаки, наприклад, атака грубої сили чи атаки MITM, можуть залишатися загрозою для безпеки мережі.

2.3 Відомі мережеві аналізатори

2.3.1 Wireshark

Wireshark, широко визнаний інструмент для аналізу мережевих пакетів, є відкритим додатком, розробленим для діагностики, моніторингу та детального вивчення мережевих протоколів, і доступний безкоштовно. Спочатку відомий як Ethereal і написаний мовами програмування C і C++, поточна стабільна версія Wireshark - 2.0.2, була випущена 26 лютого 2016 року. Одна з його відзначних функцій - це можливість кольорового кодування пакетів за протоколами, а також фільтрація та захоплення поточного трафіку в реальному часі. Також він може створювати графіки, які ілюструють дані введення/виведення та іншу статистику.

Важливо відзначити, що Wireshark дозволяє експортувати дані пакетів у різноманітні формати файлів. Зручний графічний інтерфейс користувача (GUI) спрощує аналіз пакетів, сприяючи широкому використанню цього інструмента. Крім того, програма підтримує використання через інтерфейс командного рядка. Здатність обробляти понад тисячу протоколів робить Wireshark привабливим вибором для аналізу протоколів. Нижче наведено приклад графічного інтерфейсу користувача Wireshark.

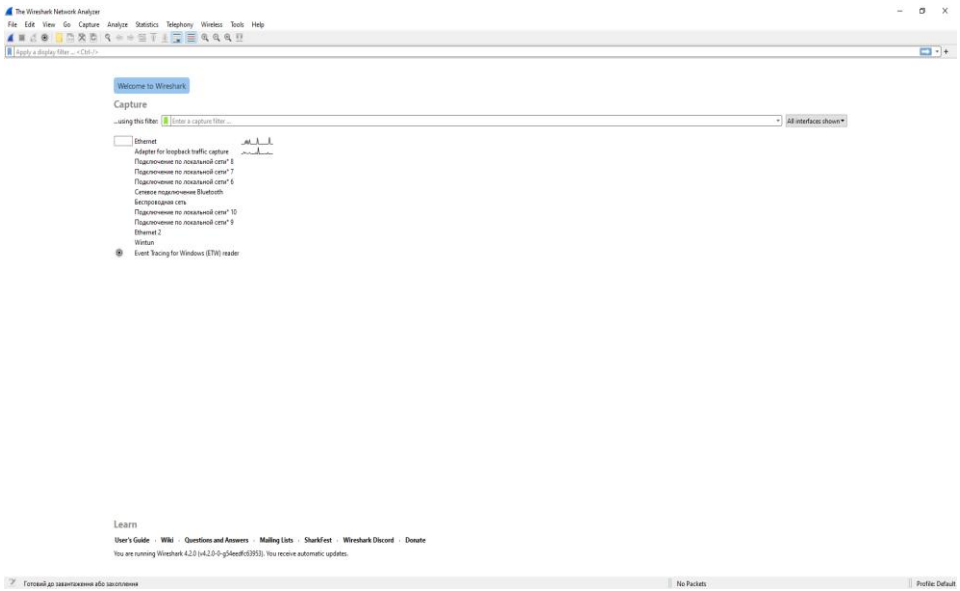


Рис. 2.1 Графічний інтерфейс користувача Wireshark

2.3.2 Tcpdump

Tcpdump - це ще один визнаний мережевий аналізатор, який працює у текстовому інтерфейсі командного рядка. Його можливості включають перегляд ідентифікаторів мережі, аналіз поведінки мережі, перегляд паролів, веб-сайтів та їхнього вмісту, який відвідує користувач. Подібно до Wireshark, tcpdump є багатоплатформовим і доступним для використання безкоштовно.

Проте для використання tcpdump користувачеві потрібні права адміністратора. Важливо відзначити, що tcpdump не такий інтуїтивний та простий у використанні, як Wireshark. Для звичайного користувача може бути важко розібратися з командним рядком, оскільки він опосередковано взаємодіє з обмеженою кількістю протоколів.

2.4 Загальні рішення методів періодизації трафіку Wireshark

2.4.1 Інсталяція Wireshark

Щоб почати користуватися програмним забезпеченням Wireshark, користувачеві необхідно його встановити. Ця програма відома своїми перевагами в галузі аналізу пакетів, такими як підтримка понад 1000 протоколів, сумісність з усіма основними операційними системами та зручний графічний інтерфейс. Однак для успішного встановлення Wireshark повинні бути виконані певні системні вимоги:

- Процесор мінімум 400 МГц.
- Мінімум 128 МБ оперативної пам'яті.
- Мінімум 75 МБ вільного місця на жорсткому диску.
- Підтримка мережевого адаптера в режимі прослуховування.
- Встановлений драйвер захоплення WinPcap.

Wireshark можна завантажити з офіційного сайту http://www.wireshark.org для операційних систем Windows, Mac OS X та Linux. На сторінці завантаження представлений інсталятор для Windows,

DMG для Mac OS і вихідний код для Linux. Для платформи Windows користувач повинен завантажити файл .exe і завершити процес інсталяції, переконавшись, що драйвер захоплення WinPcap присутній. У разі систем Linux користувач може вибирати між завантаженням вихідного коду з офіційного сайту та встановленням через системний ресурс за допомогою команди apt-get install Wireshark для дистрибутивів, подібних до Debian. Для інших дистрибутивів ви можете скомпілювати програму з вихідного коду, наведені нижче етапи компіляції:

1. Завантажте вихідний код з офіційного веб-сайту.
2. Розпакуйте архів за допомогою команди tar –jxvf downloaded_filename-version.tar.bz2.

2.4.2 Захоплення даних

Після успішної інсталяції Wireshark, перший крок - захоплення та аналіз пакетів мережі. Нижче подані етапи збору, ініціювання та отримання даних:

Відкрийте Wireshark (запустіть його через оболонку або менеджер вікон).

Користувач починає захоплювати пакети, відкривши вікно захоплення. У вікні запуску відображаються всі доступні інтерфейси мережі, які можна вибрати простим двічі клацанням на активному інтерфейсі, щоб розпочати процес захоплення.

Користувач також може ініціювати захоплення, перейшовши в меню захоплення та вибравши параметри зі спадного меню захоплення. У вікні інтерфейсів захоплення відображаються доступні інтерфейси, і користувач може розпочати перше захоплення, натиснувши на інтерфейс, в якому він/вона бажає провести захоплення. Коли Wireshark готовий захопити необхідну кількість даних, користувач може зупинити процес, натиснувши кнопку "зупинити" у спадному меню захоплення.

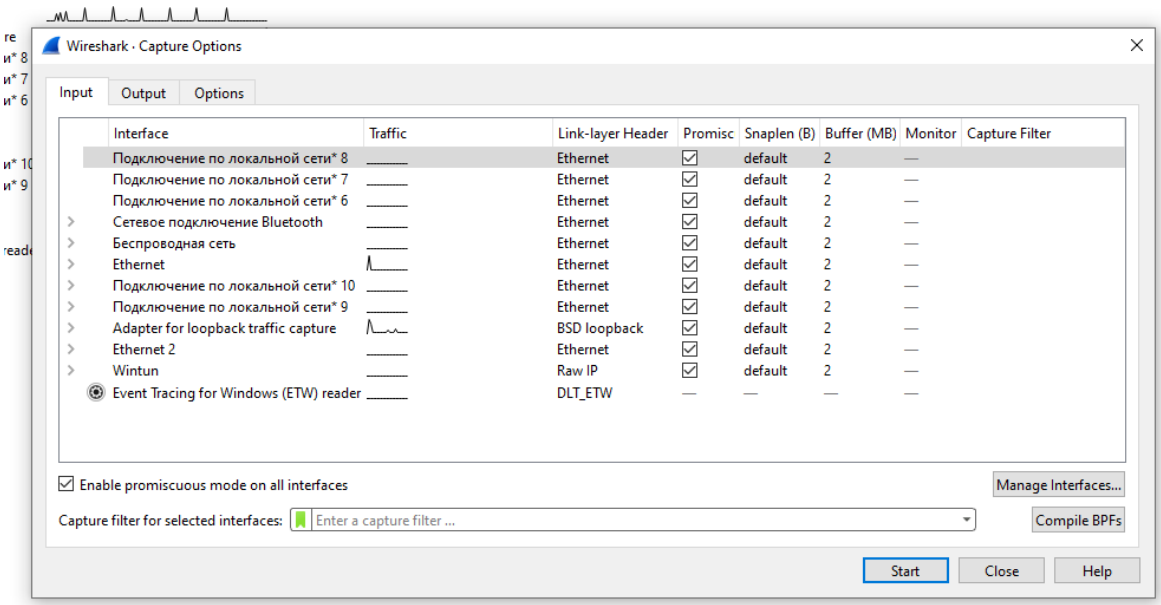


Рис. 2.2. Вибір інтерфейсу для початка захоплення пакетів

Щоб забезпечити користувачеві легкість та зручність виконання завдань, Wireshark використовує різні розділи з рівномірною кількістю ключових функцій у своєму графічному інтерфейсі. На рисунку 2.2 демонструється різні секції користувацького інтерфейсу після захоплення деяких пакетів.

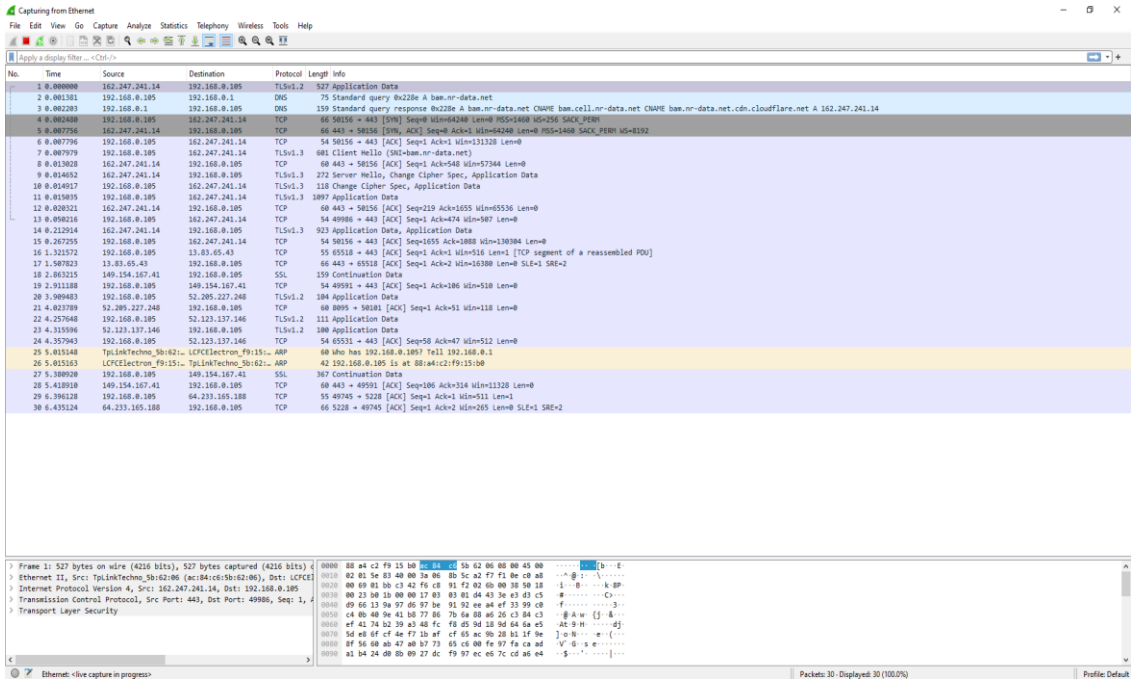


Рис. 2.3 Головне вікно після захоплення пакетів

Частини, з яких складається основне вікно Wireshark:

- Меню. Знаходиться у верхній частині основного вікна. Головні пункти зображено на рисунку 2.4 нижче:

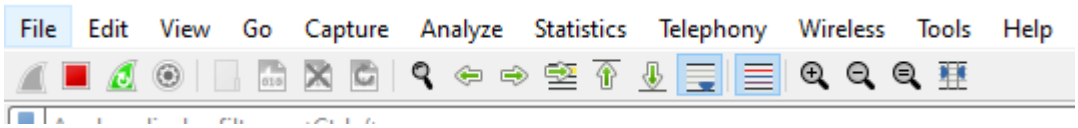


Рис. 2.4 Меню аналізатора Wireshark

- File. Меню включає різні опції, такі як друкування, відкриття, збереження та об'єднання захоплених файлів, а також експорт захоплених даних. Тут також доступна опція виходу з програми Wireshark.
- Edit. Це меню містить опції для позначення та скасування позначення пакетів.
- View. Включає функції відображення панелі інструментів і керує забарвленням та правилами для пакетів.
- Capture. Керує функцією захоплення, дозволяє користувачеві запускати та зупиняти процес. Тут також доступні налаштування для інтерфейсів захоплення та фільтрів.
- Analyze. Включає опції для плагінів, фільтрів відображення та функцій дисектора для аналізу конкретних протоколів, таких як TCP, UDP і надає експертну інформацію.
- Statistics. Меню надає графіки вводу-виводу та інші статистичні дані для різних протоколів, таких як HTTP, DNS, а також інформацію про потік та інше.
- Telephony. Містить опції для відображення вікон статистики, що стосуються телефонії, включаючи інформацію про VOIP-дзвінки та аналіз пакетів GSM через графіки для телефонних потоків.

- Wireless. Містить пункти для надання статистики трафіку Bluetooth та бездротових мереж.
- Tools. Меню включає опції для написання дисекторів за допомогою програмування Lua для Wireshark, а також для створення різних правил, таких як списки контролю доступу брандмауера. Наведення курсору на елементи надає додаткову інформацію.

Панель інструментів фільтра сприяє застосуванню та редагуванню фільтрів відображення через введення та редагування рядка фільтра. Приміром може бути панель інструментів фільтра, зображена на рисунку 2.4.

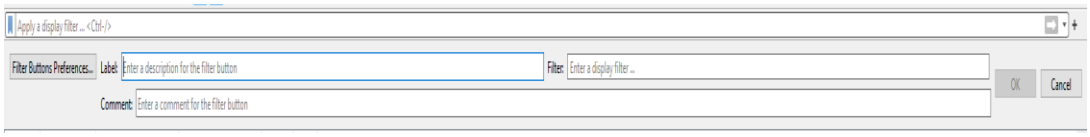
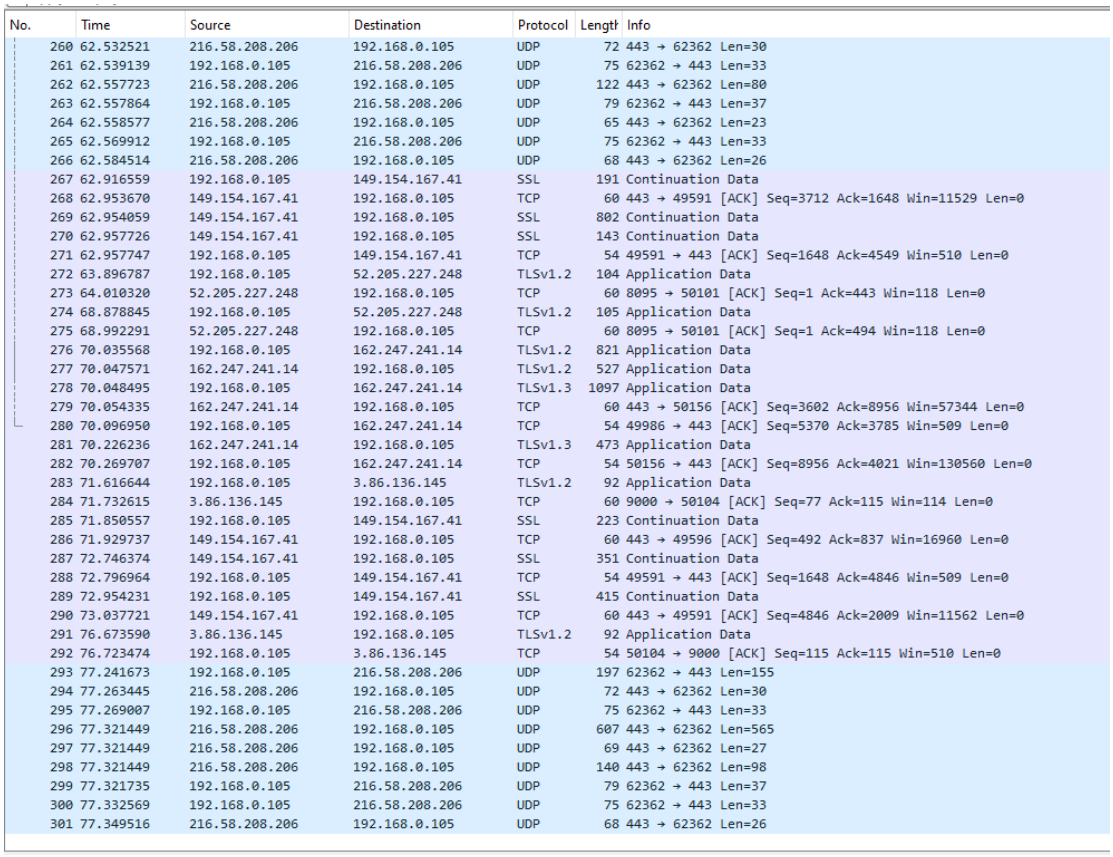


Рис. 2.5 Панель інструментів фільтра захвату

Панель списку пакетів є ключовим компонентом головного вікна Wireshark. У цьому списку відображаються всі пакети під час певного сеансу. У цьому списку кожен пакет представлений окремим рядком. Вибравши певний рядок у вікні списку пакетів і перейшовши на панелі «Відомості про пакет» та «Байти пакета», користувач може отримати докладну інформацію про вибраний пакет.

Розширена інформація про пакети представлена в різних стовпцях на панелі. Ця інформація включає додаткові дані про вміст пакета, використовуваний протокол, довжину кожного пакета, адресу призначення, адресу джерела, час витрат та кількість захоплених пакетів. Область списку пакетів із зазначеними стовпцями представлена на рисунку 2.5.



No.	Time	Source	Destination	Protocol	Length	Info
260	62.532521	216.58.208.206	192.168.0.105	UDP	72	443 → 62362 Len=30
261	62.539139	192.168.0.105	216.58.208.206	UDP	75	62362 → 443 Len=33
262	62.557723	216.58.208.206	192.168.0.105	UDP	122	443 → 62362 Len=80
263	62.557864	192.168.0.105	216.58.208.206	UDP	79	62362 → 443 Len=37
264	62.558577	216.58.208.206	192.168.0.105	UDP	65	443 → 62362 Len=23
265	62.569912	192.168.0.105	216.58.208.206	UDP	75	62362 → 443 Len=33
266	62.584514	216.58.208.206	192.168.0.105	UDP	68	443 → 62362 Len=26
267	62.916559	192.168.0.105	149.154.167.41	SSL	191	Continuation Data
268	62.953670	149.154.167.41	192.168.0.105	TCP	60	443 → 49591 [ACK] Seq=3712 Ack=1648 Win=11529 Len=0
269	62.954059	149.154.167.41	192.168.0.105	SSL	802	Continuation Data
270	62.957726	149.154.167.41	192.168.0.105	SSL	143	Continuation Data
271	62.957747	192.168.0.105	149.154.167.41	TCP	54	49591 → 443 [ACK] Seq=1648 Ack=4549 Win=510 Len=0
272	63.896787	192.168.0.105	52.205.227.248	TLSv1.2	104	Application Data
273	64.010320	52.205.227.248	192.168.0.105	TCP	60	8095 → 50101 [ACK] Seq=1 Ack=443 Win=118 Len=0
274	68.878845	192.168.0.105	52.205.227.248	TLSv1.2	105	Application Data
275	68.992291	52.205.227.248	192.168.0.105	TCP	60	8095 → 50101 [ACK] Seq=1 Ack=494 Win=118 Len=0
276	70.035568	192.168.0.105	162.247.241.14	TLSv1.2	821	Application Data
277	70.047571	162.247.241.14	192.168.0.105	TLSv1.2	527	Application Data
278	70.048495	192.168.0.105	162.247.241.14	TLSv1.3	1097	Application Data
279	70.054335	162.247.241.14	192.168.0.105	TCP	60	443 → 50156 [ACK] Seq=3602 Ack=8956 Win=57344 Len=0
280	70.096950	192.168.0.105	162.247.241.14	TCP	54	49986 → 443 [ACK] Seq=5370 Ack=3785 Win=509 Len=0
281	70.226236	162.247.241.14	192.168.0.105	TLSv1.3	473	Application Data
282	70.269707	192.168.0.105	162.247.241.14	TCP	54	50156 → 443 [ACK] Seq=8956 Ack=4021 Win=130560 Len=0
283	71.616644	192.168.0.105	3.86.136.145	TLSv1.2	92	Application Data
284	71.732615	3.86.136.145	192.168.0.105	TCP	60	9000 → 50104 [ACK] Seq=77 Ack=115 Win=114 Len=0
285	71.850557	192.168.0.105	149.154.167.41	SSL	223	Continuation Data
286	71.929737	149.154.167.41	192.168.0.105	TCP	60	443 → 49596 [ACK] Seq=492 Ack=837 Win=16960 Len=0
287	72.746374	149.154.167.41	192.168.0.105	SSL	351	Continuation Data
288	72.796964	192.168.0.105	149.154.167.41	TCP	54	49591 → 443 [ACK] Seq=1648 Ack=4846 Win=509 Len=0
289	72.954231	192.168.0.105	149.154.167.41	SSL	415	Continuation Data
290	73.037721	149.154.167.41	192.168.0.105	TCP	60	443 → 49591 [ACK] Seq=4846 Ack=2009 Win=11562 Len=0
291	76.673590	3.86.136.145	192.168.0.105	TLSv1.2	92	Application Data
292	76.723474	192.168.0.105	3.86.136.145	TCP	54	50104 → 9000 [ACK] Seq=115 Ack=115 Win=510 Len=0
293	77.241673	192.168.0.105	216.58.208.206	UDP	197	62362 → 443 Len=155
294	77.263445	216.58.208.206	192.168.0.105	UDP	72	443 → 62362 Len=30
295	77.269007	192.168.0.105	216.58.208.206	UDP	75	62362 → 443 Len=33
296	77.321449	216.58.208.206	192.168.0.105	UDP	607	443 → 62362 Len=565
297	77.321449	216.58.208.206	192.168.0.105	UDP	69	443 → 62362 Len=27
298	77.321449	216.58.208.206	192.168.0.105	UDP	140	443 → 62362 Len=98
299	77.321735	192.168.0.105	216.58.208.206	UDP	79	62362 → 443 Len=37
300	77.332569	192.168.0.105	216.58.208.206	UDP	75	62362 → 443 Len=33
301	77.349516	216.58.208.206	192.168.0.105	UDP	68	443 → 62362 Len=26

Рис. 2.6 Область списку пакетів

В області деталей пакета можна отримати більш детальну інформацію, якщо вибрати конкретний пакет у списку пакетів. У цій області представлена інформація про протоколи та поля вибраного пакета. Поля та протоколи пакета відображаються у вигляді ієрархії дерева, яку можна розгортати та згортати. На рисунку 2.6 показано приклад цієї області деталей пакета.

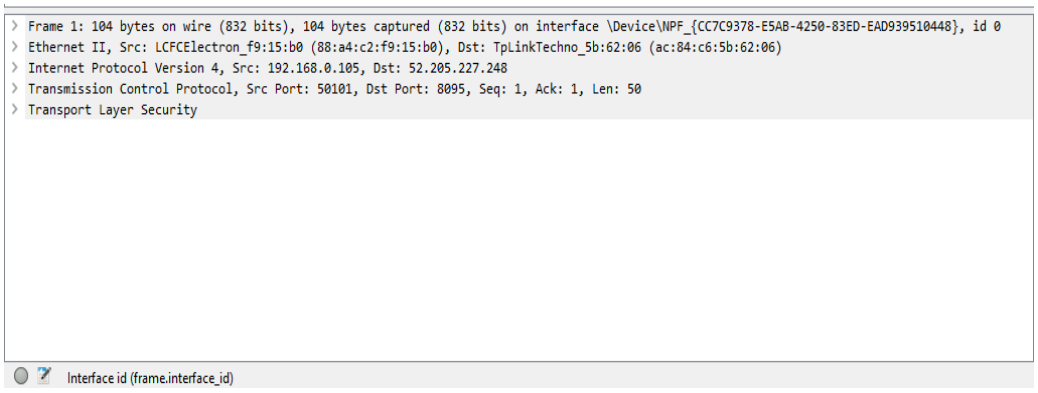


Рис. 2.7 Область деталей пакета

На рисунку 2.8 показано наступну інформацію, що відображає статус рідка, коли виконується будь-яке захоплення чи завантажується файл захоплення.

Рис 2.8 Рядок стану після завантаження захоплення

У рядку стану, розташованому поруч із ним у лівому нижньому куті, відображається ключова інформація від експерта — назва захопленого файлу. Також, на правій стороні рядка стану вказано кількість захоплених пакетів, а в правому нижньому куті відведено місце для конфігураційного профілю, який використовується для збору даних.

2.8. Дослідження роботи ДНС сервера створеного на віртуальній машині.

2.8.1 Створення віртуальної машини на базі Windows Server 2012 у середі Oracle VM Virtualbox

Після завантаження та встановлення Oracle VM Virtualbox нас зустрічає вікно «Менеджер», де ми починаємо створення віртуальної машини натисканням кнопки «Створити».

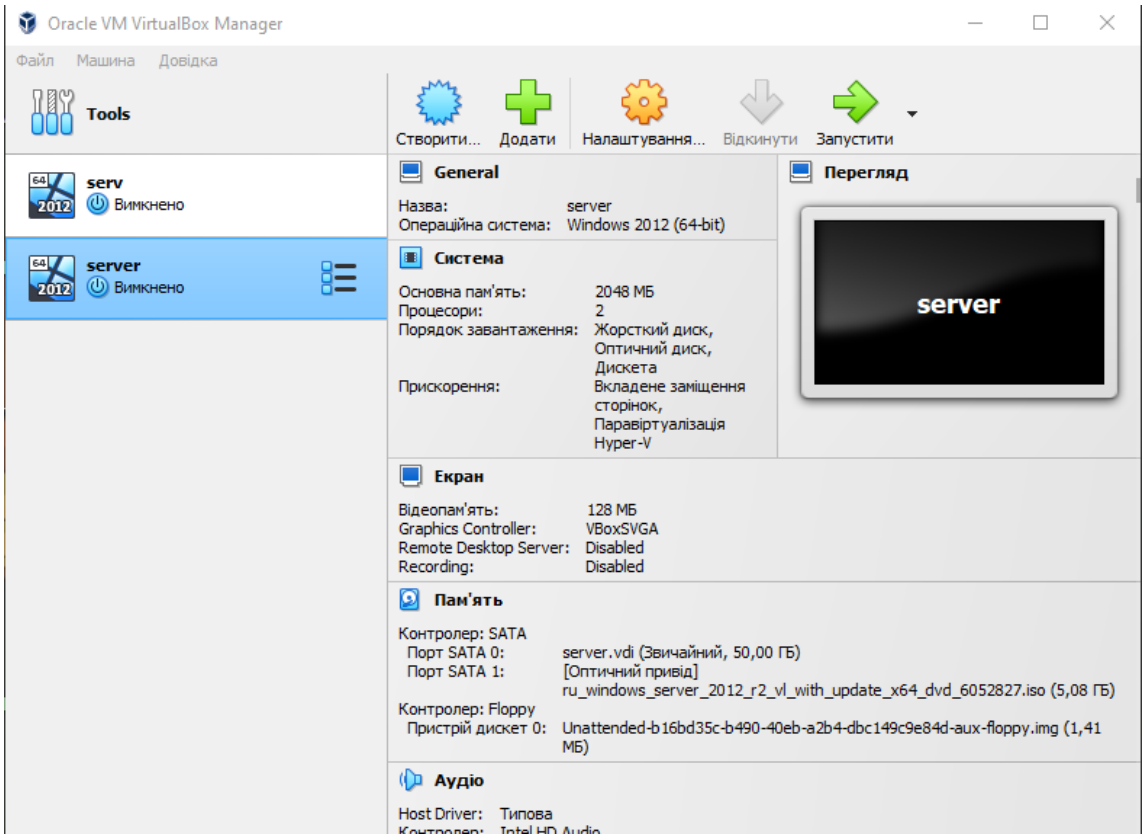


Рис. 2.9 Інтерфейс Oracle VM Virtualbox

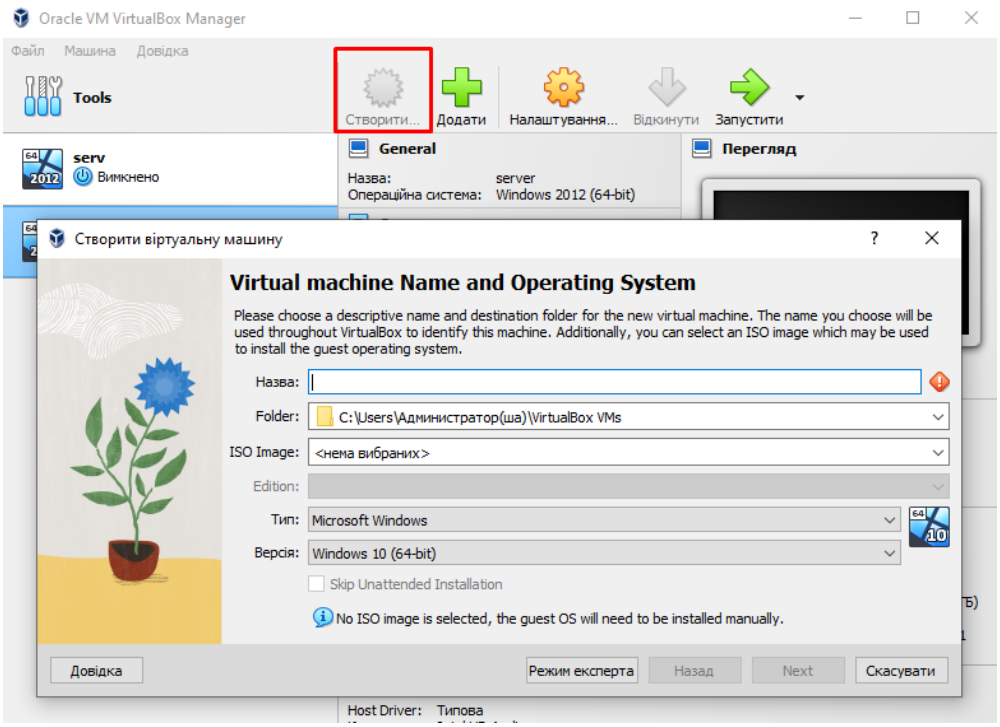


Рис 2.10 Створення віртуальної машини на базі Windows Server 2012

У вікні створення ми задаємо ім’я для ВМ та обираємо образ диску. Наступним етапом буде налаштування відштовхуючись від характеристик власного ПК.

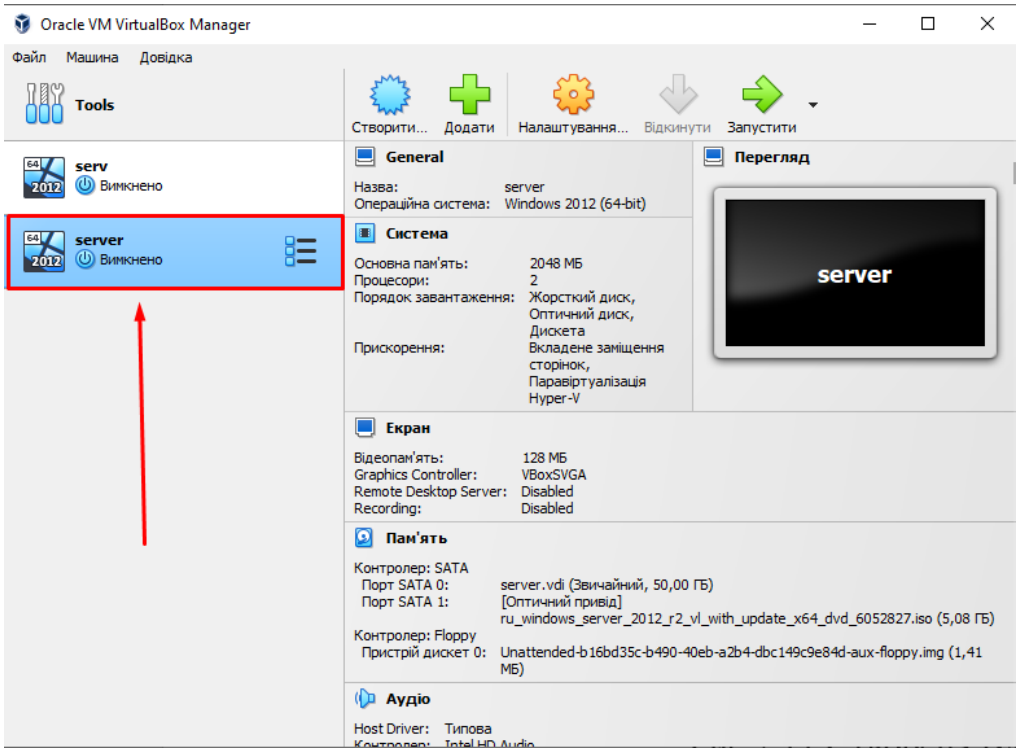


Рис.2.11 Створена ВМ з ім’ям «server»

Отримавши створену ВМ з ім’ям «server» натискаємо «Запустити» у верхньому правому кутку вікна і чекаємо завершення інсталяції операційної системи.

2.8.2 Реєстрація DNS-сервера

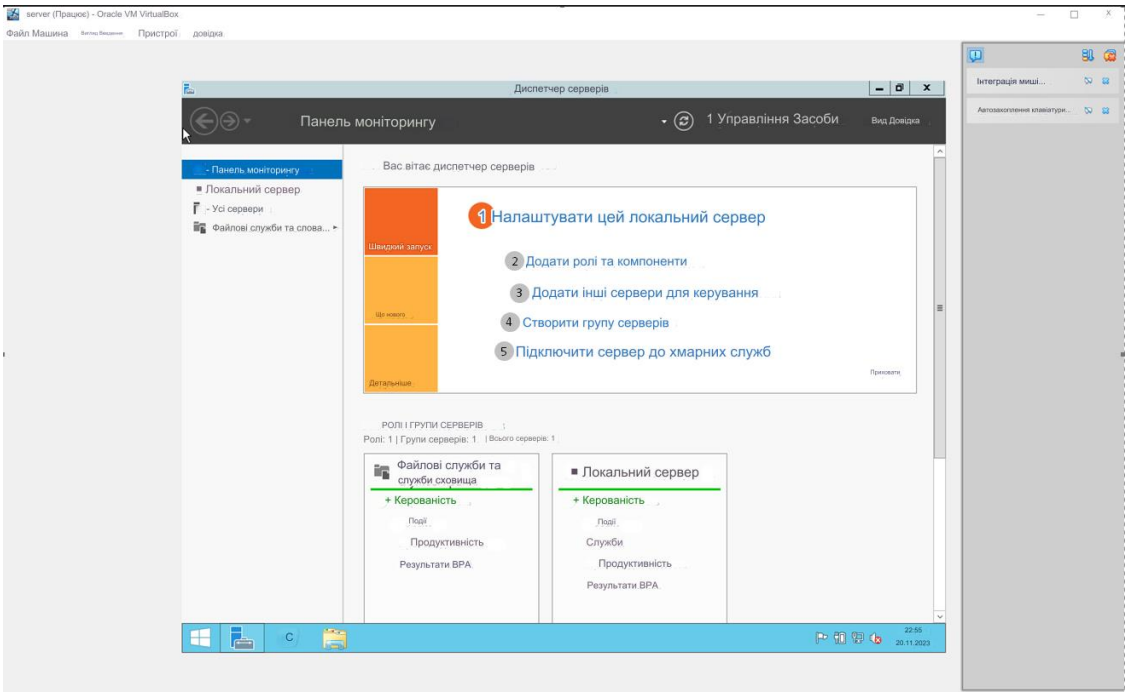


Рис.2.12 Зустрічаюче вікно під час першого запуску ОС

Починаємо реєстрацію DNS-сервера натисканням «Додати ролі і компоненти». Але перед початком нам потрібно створити статичну IP-адресу у «Параметрах мережевих підключень».

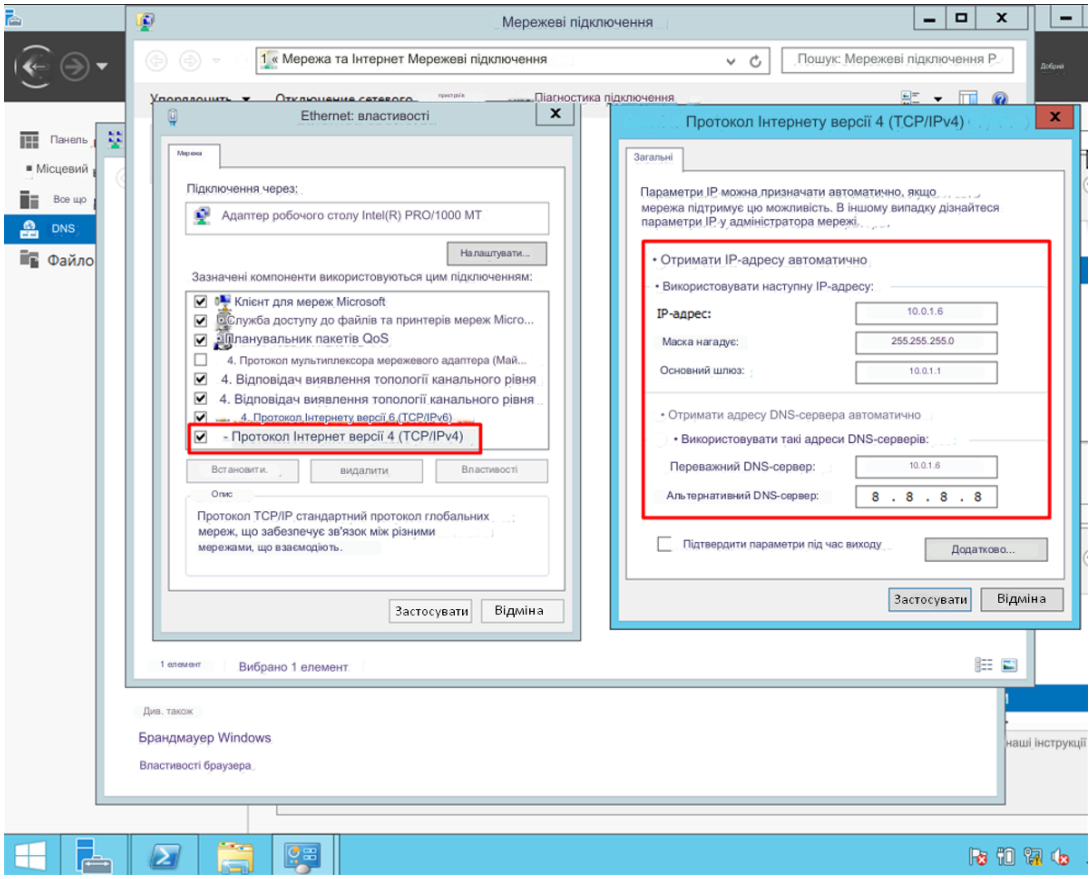


Рис. 2.13 Створення статичної IP-адреси для ВМ

Повернувшись до вікна «Майстер додавання ролей і компонентів» на етапі «Ролі сервера» ставимо галочку навпроти «DNS-сервер» і тиснемо «Далі» доки не дійдемо до етапу «Підтвердження». Тиснемо «Встановити» і чекаємо завершення.

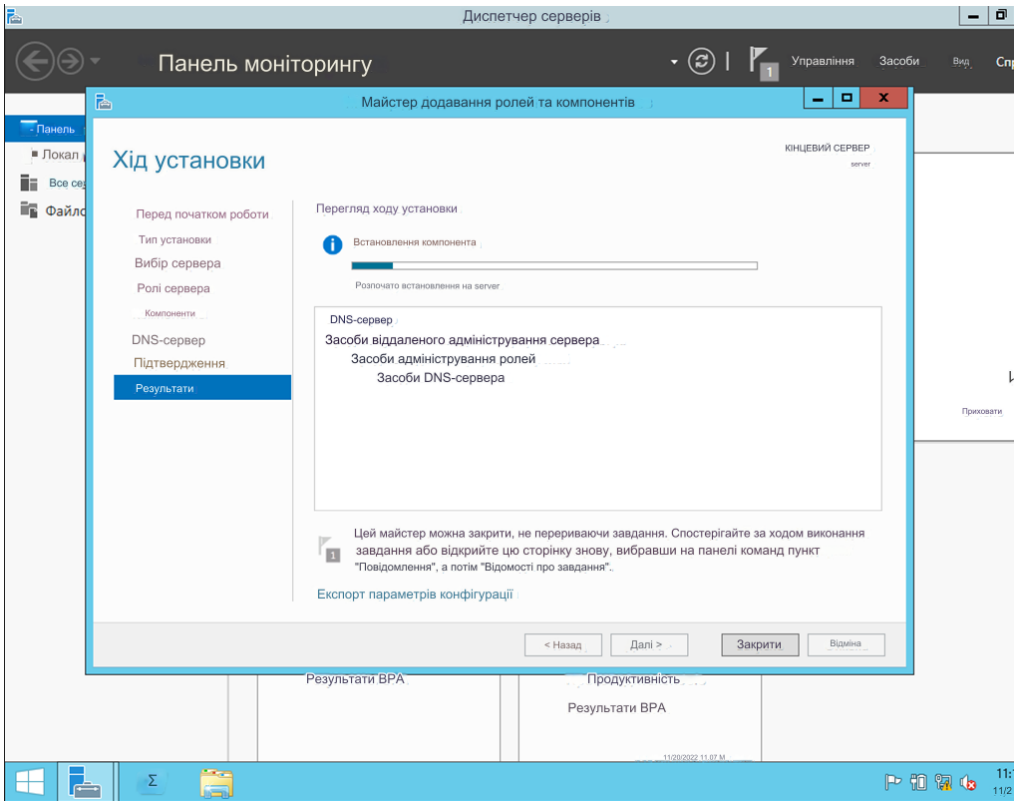


Рис.2.14 Процес інсталяції серверу

Завершенням налаштування буде етап створення зон «Прямого перегляду» та «Зворотного перегляду».

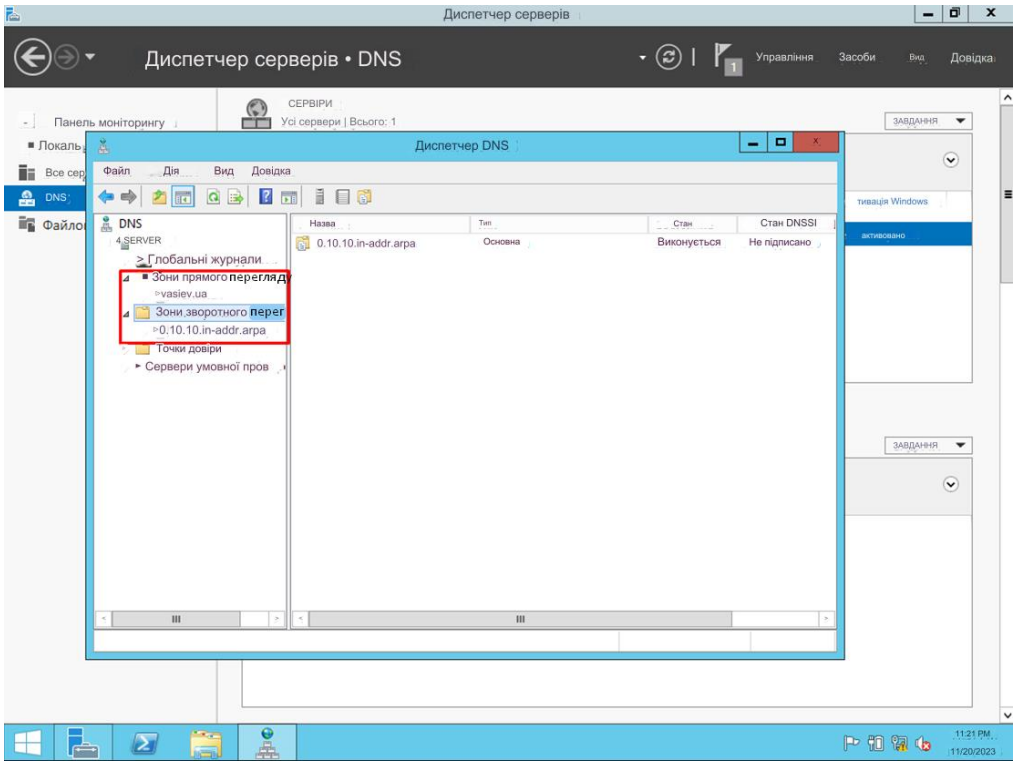


Рис.2.15 Налаштовані зони «Прямого» та «Зворотного» переглядів

2.8.3 Перевірка підключення до DNS-сервера

Для перевірки підключення клієнту (ПК) до серверу (ВМ) спочатку використовуємо Windows Powershell введенням IP-адреси серверу з командою «ssh». В нашому випадку це «ssh 10.0.1.6»

Рис.2.16 Перевірка підключення командою «ssh»

Переходимо до програми Wireshark та виконуємо «захват» мережі Ethernet, до якої підключені і сервер і клієнт.

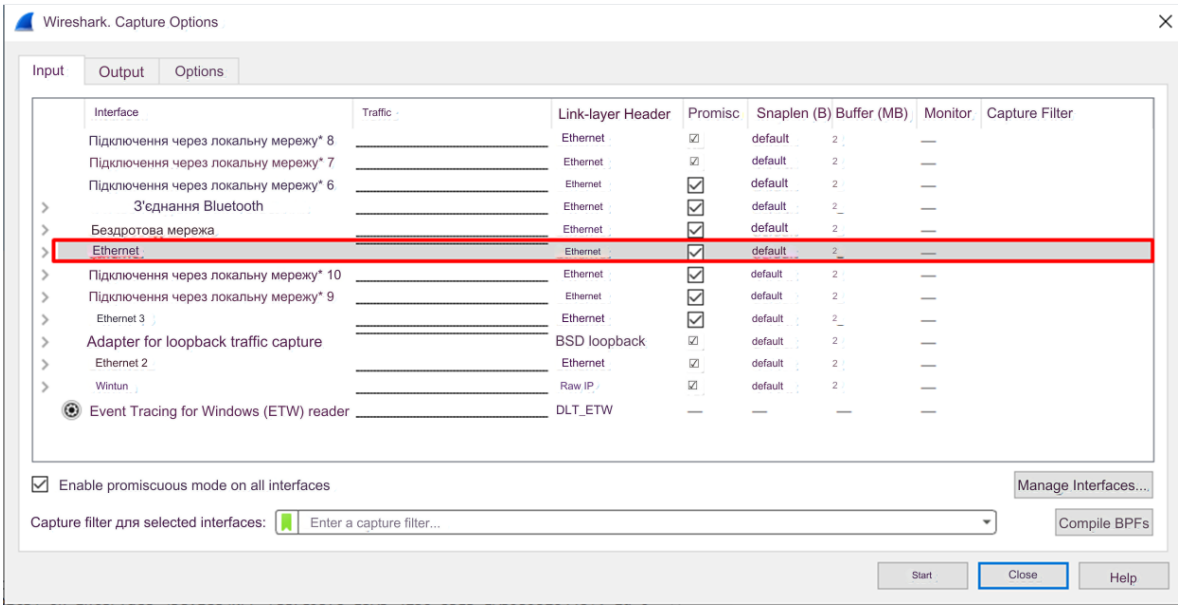


Рис.2.17 «захват» мережі Ethernet

Для фільтрації протоколу DNS вводимо у рядок «current filter» значення «udp.port == 53».

Далі знаходимо підключення, яке ми виконували вище, у даному списку.

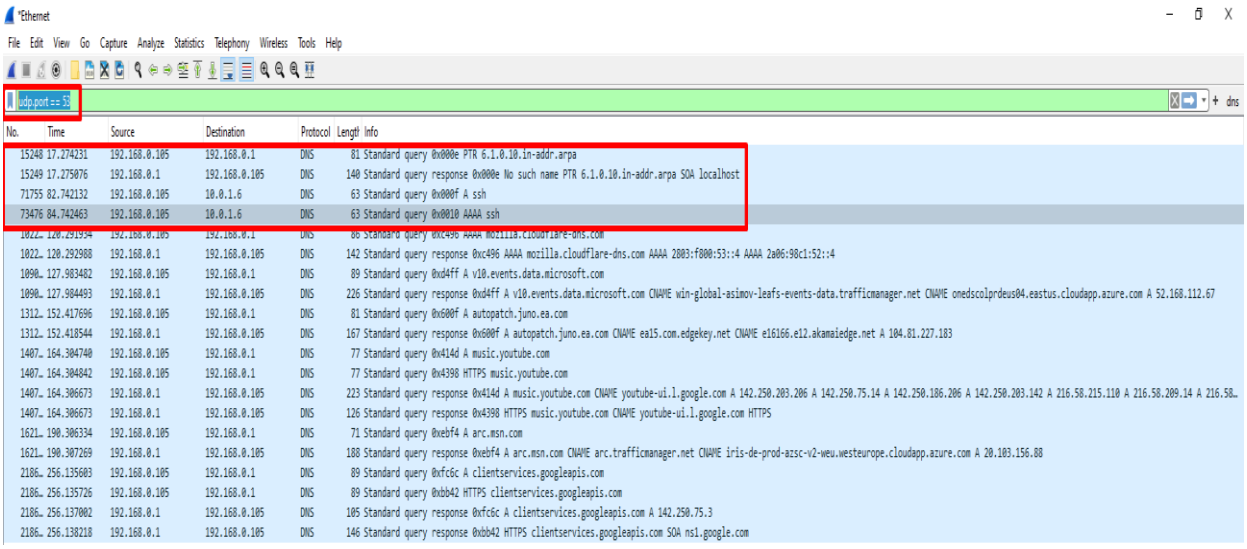


Рис.2.18 Результат фільтрації протоколів

Разом отримуємо очікуваний аналіз підключення клієнта (ПК) до DNS-сервера (ВМ).

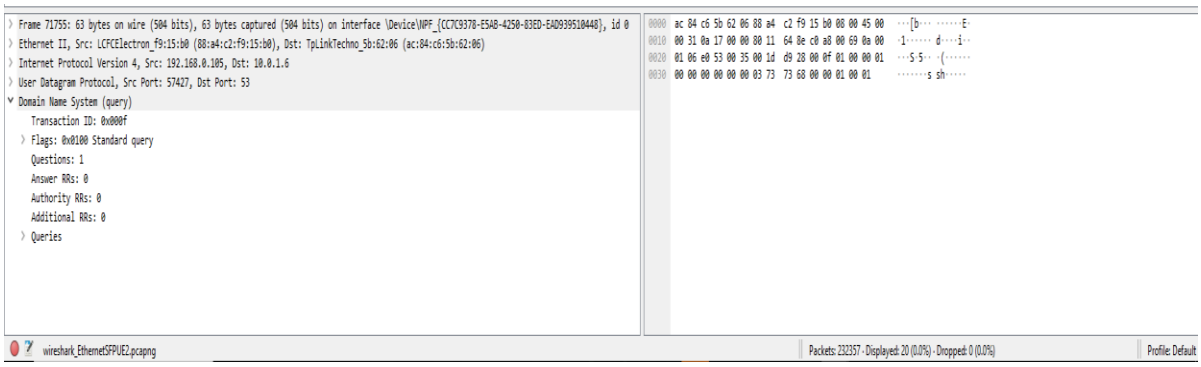


Рис.2.19 Аналіз підключення клієнта до DNS-сервера

Висновки за розділом

Дослідження процедур обміну повідомлень в системі DNS виявило ряд ключових аспектів, які визначають ефективність та надійність цієї служби в мережевому середовищі. З'ясовано, що процес обміну повідомлень у DNS ґрунтується на чітко визначених протоколах та стандартах, зокрема на принципах роботи протоколу DNS, який визначає взаємодію між різними складовими системи.

Мережеві аналізатори, як ключовий інструмент у сфері інформаційних технологій, виявляються невід'ємною частиною адміністрування та моніторингу комп'ютерних мереж. Дослідження їх функцій та можливостей підтверджує, що ці засоби грають важливу роль у виявленні, аналізі та вирішенні проблем, пов'язаних із трафіком, безпекою та ефективністю мережевого взаємодії.

Одним із ключових висновків є те, що мережеві аналізатори забезпечують операторам мережі високий рівень видимості в процесах передачі даних, дозволяючи вчасно виявляти і вирішувати проблеми з підтримкою стабільності та продуктивності мережі.

РОЗДІЛ 3
РОЗРАХУНКИ ПАРАМЕТРІВ DNS СЕРВЕРУ ТА ЗАПИТІВ ДО НИХ

3.1 Розрахунок кількості запитів DNS

Кількість запитів DNS може бути розрахована за наступною формулою:

$Q = N * R$

де:

Q - кількість запитів DNS

N - кількість кінцевих користувачів

R - середня кількість запитів на одного кінцевого користувача

Давайте розглянемо сценарій, де в мережі присутні 1000 кінцевих користувачів, і в середньому кожен кінцевий користувач генерує 10 запитів DNS на день. Відповідно, загальна кількість запитів DNS розраховується наступним чином: Q=N×R=1000×10=10000Q=N×R=1000×10=10000 Отже, в мережі щоденно відбувається 10000 запитів DNS.

Тепер давайте оцінимо можливе збільшення запитів DNS, якщо кількість кінцевих користувачів зросте до 10000. Підставивши нову кількість кінцевих користувачів у формулу, ми отримуємо: Q=N×R=10000×10=100000Q=N×R=10000×10=100000 Це означає, що при 10000 кінцевих користувачах в мережі щоденно відбувається 100000 запитів DNS.

Зі збільшенням кількості кінцевих користувачів збільшується і кількість запитів DNS, оскільки кожен кінцевий користувач ініціює більше запитів DNS.

Розгляд навантаження на DNS-сервер: Припустимо, що кількість запитів DNS становить 10000 запитів на день, а середній розмір запиту DNS складає 100 байтів. Тоді навантаження на DNS-сервер розраховується наступним чином: L=Q×S=10000×100=1000000L=Q×S=10000×100=1000000 Це означає, що DNS-сервер щоденно обробляє 1000000 байтів даних.

Тепер давайте оцінимо вплив збільшення середнього розміру запиту DNS до 200 байтів. Підставивши новий середній розмір у формулу, ми отримуємо: L=Q×S=10000×200=2000000L=Q×S=10000×200=2000000 Це означає, що навантаження на DNS-сервер збільшиться до 2000000 байтів даних.

Як видно, зі збільшенням середнього розміру запиту DNS навантаження на DNS-сервер також збільшується. Це пов'язано з тим, що DNS-сервер повинен обробляти більше даних для кожного окремого запиту.

3.2 Навантаження на DNS-сервер

Параметр	Опис
Кількість запитів на хвилину	Кількість запитів, які надходять на DNS-сервер за хвилину.
Середній час на запит	Середній час, який витрачається на обробку одного запиту.
Ширина смуги пропускання	Ширина смуги пропускання, яка доступна DNS-серверу.

Рисунок 1.1 – Визначення кількості запитів на DNS-сервер за певний період часу

Для розрахунку навантаження на DNS-сервер необхідно знати два значення:

- Кількість запитів DNS
- Середній розмір запиту DNS

3.2.1 Кількість запитів DNS

Кількість запитів DNS - це кількість запитів DNS, які обробляє DNS-сервер за певний період часу.

Середній розмір запиту DNS - це розмір одного запиту DNS.

Приклад розрахунку навантаження на DNS-сервер

Припустимо, що кількість запитів DNS становить 10000 запитів на день, а середній розмір запиту DNS становить 100 байт. Тоді навантаження на DNS-сервер складе:

$L = Q * S = 10000 * 100 = 1000000$

Це означає, що щодня DNS-сервер обробляє 1000000 байт даних.

Розрахунок навантаження на DNS-сервер на основі статистики

Якщо у вас є доступ до статистики запитів DNS, то ви можете використовувати її для розрахунку навантаження на DNS-сервер.

Статистика запитів DNS може містити такі дані:

- Кількість запитів DNS за певний період часу
- Типи запитів DNS
- Домени, на які були здійснені запити

Використовуючи ці дані, ви можете розрахувати навантаження на DNS-сервер за певний період часу, а також розподіл навантаження по типах і доменам.

Розрахунок навантаження на DNS-сервер на основі прогнозу

Якщо ви хочете спрогнозувати навантаження на DNS-сервер в майбутньому, ви можете використовувати наступні фактори:

- Очікуване зростання кількості запитів DNS
- Очікуване зростання середнього розміру запиту DNS

Використовуючи ці фактори, ви можете скласти прогноз навантаження на DNS-сервер на певний період часу.

Додаткові фактори, які можуть впливати на навантаження на DNS-сервер

Навантаження на DNS-сервер може також впливати ряд інших факторів, таких як:

- Типи запитів DNS
- Домени, на які були здійснені запити
- Час доби
- День тижня

Враховуючи ці фактори, ви можете отримати більш точний прогноз навантаження на DNS-сервер.

3.2.2 Розмір повідомлення DNS

Розмір повідомлення DNS можна знайти за наступною формулою:

*Розмір = 12 + (Кількість ресурсів * Розмір ресурсу)*

де:

Розмір - розмір повідомлення DNS

12 - розмір заголовка повідомлення DNS

Кількість ресурсів - кількість ресурсів, які містяться в тілі повідомлення

Розмір ресурсу - розмір одного ресурсу

Приклад розрахунку розміру повідомлення DNS

Припустимо, що в тілі повідомлення DNS міститься 10 ресурсів, кожен з яких має розмір 20 байт. Тоді розмір повідомлення DNS складе:

*Розмір = 12 + (10 * 20) = 222*

Це означає, що повідомлення DNS буде мати розмір 222 байта.

Розрахунок кількості ресурсів в повідомленні DNS

Кількість ресурсів в повідомленні DNS можна розрахувати за наступною формулою:

Кількість ресурсів = (Розмір - 12) / Розмір ресурсу

де:

Кількість ресурсів - кількість ресурсів, які містяться в тілі повідомлення

Розмір - розмір повідомлення DNS

12 - розмір заголовка повідомлення DNS

Розмір ресурсу - розмір одного ресурсу

Приклад розрахунку кількості ресурсів в повідомленні DNS

Припустимо, що розмір повідомлення DNS становить 222 байта, а розмір одного ресурсу становить 20 байт. Тоді кількість ресурсів в повідомленні DNS складе:

Кількість ресурсів = (222 - 12) / 20 = 10

Це означає, що в тілі повідомлення DNS міститься 10 ресурсів.

Додаткові розрахунки

Розрахунок часу, необхідного для передачі повідомлення DNS - час, необхідне для передачі повідомлення DNS, залежить від розміру повідомлення і пропускної здатності каналу зв'язку.

Розрахунок витрат на передачу повідомлення DNS - витрати на передачу повідомлення DNS залежать від розміру повідомлення і вартості трафіку.

Ці розрахунки можуть бути корисні для планування та оптимізації роботи системи доменних імен.

3.3 Розрахунок часу, необхідного для отримання відповіді DNS

Час, необхідне для отримання відповіді DNS, залежить від таких факторів:

Навантаження на DNS-сервер - час, необхідне для обробки запиту DNS, збільшується зі збільшенням навантаження на DNS-сервер.

Відстань між DNS-клієнтом і DNS-сервером - час, необхідний для передачі запиту DNS і відповіді DNS, збільшується зі збільшенням відстані між DNS-клієнтом і DNS-сервером.

Час, необхідний для отримання відповіді DNS

Припустимо, що навантаження на DNS-сервер становить 10000 запитів на секунду, а відстань між DNS-клієнтом і DNS-сервером становить 1000 км. Тоді час, необхідне для отримання відповіді DNS, складе:

*Час = (1 / 10000) * 1000 * 8 / 1000000000*

де:

Час - час, необхідний для отримання відповіді DNS

1 / 10000 - середня тривалість обробки одного запиту DNS

1000 - відстань між DNS-клієнтом і DNS-сервером в кілометрах

8 - швидкість передачі даних по оптоволоконному кабелю в мегабітах на секунду

1000000000 - 1 мегабіт в кілобітах

У цьому випадку час, необхідне для отримання відповіді DNS, складе:

*Час = (1 / 10000) * 1000 * 8 / 1000000000 = 0,00008* секунди

Це означає, що відповідь DNS буде отримана через 0,00008 секунди.

Додаткові розрахунки

Розрахунок кількості запитів DNS, які можуть обробити DNS-сервери - кількість запитів DNS, які можуть обробити DNS-сервери, залежить від навантаження на DNS-сервери і від продуктивності DNS-серверів.

Розрахунок витрат на передачу запиту DNS - витрати на передачу запиту DNS залежать від відстані між DNS-клієнтом і DNS-сервером, а також від вартості трафіку.

Ці розрахунки можуть бути корисними для планування та оптимізації роботи DNS-служби.

Заходи щодо вирішення проблем функціонування DNS-служби

Для вирішення проблем функціонування DNS-служби необхідно вжити таких заходів:

- Розширення мережі DNS-серверів - для зниження навантаження на DNS-сервери необхідно збільшити кількість DNS-серверів, розташованих у різних частинах світу. Це дозволить розподілити навантаження на більшу кількість серверів, що призведе до зниження часу, необхідного для отримання відповіді DNS.
- Розробка нових методів захисту DNS-серверів від атак - для захисту DNS-серверів від атак необхідно розробляти нові методи захисту, які будуть ускладнювати або унеможливлювати проведення атак. Ці методи можуть включати в себе використання нових протоколів безпеки, а також вдосконалення існуючих протоколів безпеки.
- Покращення стабільності DNS-системи - для підвищення стабільності DNS-системи необхідно вжити заходів щодо підвищення надійності та відмовостійкості DNS-серверів. Ці заходи можуть включати в себе використання резервних копій даних, а також впровадження нових методів виявлення та усунення неполадок.

Конкретні приклади

Ось кілька конкретних прикладів заходів, які можуть бути вжиті для вирішення проблем функціонування DNS-служби:

Розширення мережі DNS-серверів - можна збільшити кількість DNS-серверів, які належать державним або приватним організаціям. Ці сервери можуть бути розташовані в різних частинах світу, що допоможе розподілити навантаження і знизити час, необхідний для отримання відповіді DNS.

Розробка нових методів захисту DNS-серверів від атак - можна розробити нові протоколи безпеки, які будуть ускладнювати або унеможливлювати проведення атак на DNS-сервери. Ці протоколи можуть включати в себе використання нових алгоритмів шифрування, а також впровадження нових методів аутентифікації.

Покращення стабільності DNS-системи - можна впровадити нові методи виявлення і усунення неполадок, які допоможуть швидко виявити і усунути проблеми, які можуть призвести до збоїв в роботі DNS-служби.

Впровадження заходів

Впровадження заходів щодо вирішення проблем функціонування DNS-служби є складним завданням, яке потребує участі різних організацій. Державні організації можуть сприяти впровадженню заходів щодо розширення мережі DNS-серверів і розробки нових методів захисту DNS-серверів від атак. Приватні організації можуть сприяти впровадженню заходів щодо покращення стабільності DNS-системи.

Впровадження заходів щодо вирішення проблем функціонування DNS-служби має важливе значення для забезпечення стабільної роботи Інтернету.

Дослідження роботи DNS-сервера створеного на віртуальній машині

Параметр	Опис
Віртуальна машина	Віртуальна машина, на якій встановлено DNS-сервер.
Операційна система	Операційна система, на якій працює DNS-сервер.
Версія DNS-сервера	Версія DNS-сервера, який встановлений на віртуальній машині.
Метод розподілу навантаження	Метод розподілу навантаження, який використовується на DNS-сервері.
Ширина смуги пропускання	Ширина смуги пропускання, яка доступна DNS-серверу.
Кількість запитів на хвилину	Кількість запитів, які надходять на DNS-сервер за хвилину.
Середній час на запит	Середній час, який витрачається на обробку одного запиту.
Навантаження на DNS-сервер	Навантаження на DNS-сервер, яке визначається за формулою:Навантаження = Кількість запитів на хвилину * Середній час на запит

Рисунок 1.2 – Дослідження роботи DNS-сервера

Для дослідження роботи DNS-сервера, створеного на віртуальній машині, можна виконати наступні кроки:

1. Встановити DNS-сервер на віртуальну машину.
2. Налаштувати DNS-сервер.

3. Визначити метод розподілу навантаження, який використовується на DNS-сервері.
4. Визначити ширину смуги пропускання, яка доступна DNS-серверу.
5. Виміряти кількість запитів, які надходять на DNS-сервер за хвилину.
6. Виміряти середній час на запит.
7. Розрахувати навантаження на DNS-сервер.

Результати дослідження

Результати дослідження роботи DNS-сервера можна представити у вигляді таблиці. У таблиці можна вказати всі параметри, які були визначені в процесі дослідження.

Наприклад, таблиця може виглядати так:

Параметр	Значення
Віртуальна машина	Ubuntu 22.04 LTS
Операційна система	Linux
Версія DNS-сервера	Bind 9.16.1
Метод розподілу навантаження	Round-robin
Ширина смуги пропускання	100 Мбіт/с
Кількість запитів на хвилину	10000
Середній час на запит	100 мілісекунд
Навантаження на DNS-сервер	100 запитів/с

Рисунок 1.2 – Таблиця з дослідженням роботи DNS-сервера

Приклад 1.

DNS-сервер, створений на віртуальній машині, обробляє в середньому 1000 запитів на секунду. При цьому середній розмір запиту становить 100 байт. Тоді завантаження віртуальної машини складе:

*Завантаження = Кількість запитів * Розмір запиту / Пропускна здатність*

*Завантаження = 1000 * 100 / 1000000*

Завантаження = 0,1

Це означає, що завантаження віртуальної машини становить 10%.

Якщо пропускна здатність віртуальної машини становить 100 Мбіт/с, то час, необхідний для обробки одного запиту, складе:

Час = Розмір запиту / Пропускна здатність

Час = 100 / 1000000

Час = 0,0001

Це означає, що один запит обробляється за 0,0001 секунди.

Приклад 2.

Припустимо, що ми хочемо створити DNS-сервер на віртуальній машині, яка буде обробляти в середньому 1000 запитів на секунду. При цьому середній розмір запиту становить 100 байт.

Для цього нам необхідно вибрати віртуальну машину з достатньою пропускною здатністю. Виходячи з розрахунків, пропускна здатність віртуальної машини повинна становити не менше 10 Мбіт/с.

Також нам необхідно вибрати віртуальну машину з достатньою пам'яттю. Для зберігання інформації про доменні імена та їх відповідність IP-адресам нам знадобиться не менше 1 Гб оперативної пам'яті.

Нарешті, нам необхідно встановити на віртуальну машину програмне забезпечення DNS-сервера. Для цього можна використовувати, наприклад, програмне забезпечення BIND або PowerDNS.

Після того, як ми виконаємо всі ці кроки, ми зможемо протестувати роботу DNS-сервера. Для цього можна використовувати, наприклад, інструмент iperf для вимірювання продуктивності сервера.

Приклад 3.

Припустимо, що ми хочемо підвищити безпеку DNS-сервера, створеного на віртуальній машині. Для цього нам необхідно:

Ізолювати DNS-сервер від інших віртуальних машин. Це можна зробити, використовуючи функції віртуалізації, такі як віртуальні мережі або віртуальні диски.

Застосувати брандмауер для блокування несанкціонованого доступу до DNS-сервера.

Оновити програмне забезпечення DNS-сервера до останньої версії.

Після того, як ми виконаємо ці кроки, ми зможемо підвищити безпеку DNS-сервера і захистити його від атак.

3.4 Створення віртуальної машини на базі Windows Server 2012 у середі Oracle VM Virtualbox

Вибір апаратного забезпечення

Для створення віртуальної машини на базі Windows Server 2012 у середі Oracle VM Virtualbox необхідно мати наступне апаратне забезпечення:

- Комп'ютер із процесором не нижче Intel Core i3 або AMD Ryzen 3
- Оперативна пам'ять не менше 4 Гб
- Дисковий простір не менше 20 Гб

Вибір розміру віртуальної машини

Розмір віртуальної машини залежить від того, які програми та ресурси ви будете використовувати на віртуальній машині. Для установки та роботи Windows Server 2012 необхідно не менше 10 Гб дискового простору.

Вибір кількості процесорів і пам'яті для віртуальної машини

Кількість процесорів і пам'яті для віртуальної машини також залежить від того, які програми та ресурси ви будете використовувати на віртуальній машині. Для установки та роботи Windows Server 2012 можна використовувати один процесор і 2 Гб пам'яті.

Приклади:

Приклад 1

Комп'ютер із процесором Intel Core i5, 8 Гб оперативної пам'яті і 1 ТБ дискового простору. Потрібно створити віртуальну машину на базі Windows Server 2012, яка буде використовуватися для зберігання файлів і веб-сервера.

Для цього необхідно створити віртуальну машину з наступними параметрами:

- Розмір диска: 20 Гб

- Кількість процесорів: 1
- Оперативна пам'ять: 4 Гб
- Приклад 2
- Комп'ютер із процесором AMD Ryzen 7, 16 Гб оперативної пам'яті і 2 ТБ дискового простору.

Потрібно створити віртуальну машину на базі Windows Server 2012, яка буде використовуватися для віртуалізації серверів і робочих станцій.

Для цього необхідно створити віртуальну машину з наступними параметрами:

- Розмір диска: 100 Гб
- Кількість процесорів: 2
- Оперативна пам'ять: 8 Гб

Інструкція по створенню віртуальної машини на базі Windows Server 2012 у середовищі Oracle VM Virtualbox

1. Запустіть програму Oracle VM Virtualbox.
2. Натисніть кнопку "Створити".
3. У вікні "Створення нової віртуальної машини" виберіть опцію "Створити пусту віртуальну машину".
4. Уведіть ім'я та розташування віртуальної машини.
5. У полі "Система" виберіть операційну систему "Microsoft Windows".
6. У полі "Версія" виберіть версію Windows Server 2012.
7. Натисніть кнопку "Далі".
8. Виберіть розмір диска для віртуальної машини.
9. Натисніть кнопку "Створити".
10. Віртуальна машина буде створена.
11. Для установки Windows Server 2012 на віртуальну машину натисніть кнопку "Запуск".
12. Виконайте інструкцію по установці Windows Server 2012.

13. Після установки Windows Server 2012 віртуальна машина буде готова до використання.

Приклад 3

Створення віртуальної машини на базі Windows Server 2012 для використання як віртуального веб-сервера. Для цього необхідно створити віртуальну машину з наступними параметрами:

- Розмір диска: 20 Гб
- Кількість процесорів: 1
- Оперативна пам'ять: 4 Гб

Також необхідно встановити на віртуальну машину веб-сервер, наприклад, Apache або IIS.

Приклад 4

Потрібно створити віртуальну машину на базі Windows Server 2012 для використання в якості віртуальної бази даних. Для цього необхідно створити віртуальну машину з наступними параметрами:

- Розмір диска: 50 Гб
- Кількість процесорів: 2
- Оперативна пам'ять: 8 Гб

Також необхідно встановити на віртуальну машину сервер баз даних, наприклад, Microsoft SQL Server або Oracle Database.

Приклад 5

Потрібно створити віртуальну машину на базі Windows Server 2012 для використання як віртуального робочого місця. Для цього необхідно створити віртуальну машину з наступними параметрами:

- Розмір диска: 100 Гб
- Кількість процесорів: 2
- Оперативна пам'ять: 16 Гб

Також необхідно встановити на віртуальну машину потрібні програми та додатки.

Ці приклади лише ілюструють можливі варіанти створення віртуальної машини на базі Windows Server 2012 у середі Oracle VM Virtualbox. Конкретні параметри віртуальної машини залежать від потреб і завдань, які будуть виконуватися на віртуальній машині.

3.5 Реєстрація DNS-сервера

Вибір реєстратора

Першим кроком у реєстрації DNS-сервера є вибір реєстратора. Реєстратор - це компанія, яка надає послуги реєстрації доменних імен. При виборі реєстратора необхідно враховувати такі фактори:

Ціна - вартість реєстрації доменного імені залежить від реєстратора.

Функціональність - реєстратори пропонують різні набори функцій для управління доменними іменами.

Підтримка - важливо, щоб реєстратор надавав хорошу підтримку клієнтам.

Вибір доменного імені

Наступним кроком є вибір доменного імені. Доменне ім'я - це назва веб-сайту або іншого ресурсу в Інтернеті. При виборі доменного імені необхідно враховувати такі фактори:

Короткість і простота запам'ятовування - доменне ім'я повинно бути коротким і простим для запам'ятовування.

Унікальність - доменне ім'я повинно бути унікальним, щоб його не міг використовувати хтось інший.

Релевантність - доменне ім'я повинно бути релевантним до веб-сайту або іншого ресурсу, для якого воно використовується.

Реєстрація доменного імені

Після вибору реєстратора та доменного імені можна приступати до реєстрації доменного імені. Для цього необхідно виконати такі дії:

1. Зареєструйтеся на веб-сайті реєстратора.

2. Введіть доменне ім'я, яке ви хочете зареєструвати.
3. Введіть контактну інформацію для власника доменного імені.
4. Виберіть термін реєстрації доменного імені.
5. Оплатіть реєстрацію доменного імені.
6. Налаштування DNS-сервера
7. Після реєстрації доменного імені необхідно налаштувати DNS-сервер.

Для налаштування DNS-сервера необхідно виконати такі дії:

1. Виберіть DNS-сервер, який ви хочете використовувати.
2. Отримайте IP-адреси DNS-серверів.
3. Налаштуйте DNS-клієнт на вашому комп'ютері або пристрої.

Розрахунок вартості реєстрації доменного імені

Вартість реєстрації доменного імені залежить від реєстратора. В середньому вартість реєстрації доменного імені становить від 10 до 100 доларів США на рік.

Вартість налаштування DNS-сервера

Вартість налаштування DNS-сервера залежить від того, який DNS-сервер ви використовуєте. Якщо ви використовуєте DNS-сервер, який надається вашим реєстратором, то додаткових витрат не буде. Якщо ви використовуєте сторонній DNS-сервер, то вам необхідно буде придбати або орендувати сервер, а також налаштувати його.

Приклад 1

Потрібно зареєструвати доменне ім'я "example.com" на рік. Вибираємо реєстратора, який пропонує послугу реєстрації доменних імен за ціною 15 доларів США на рік. Таким чином, загальна вартість реєстрації доменного імені "example.com" складе 15 доларів США.

Приклад 2

Треба використовувати сторонній DNS-сервер, який коштує 50 доларів США на місяць. Таким чином, загальна вартість використання стороннього

DNS-сервера для доменного імені "example.com" складе 600 доларів США на рік.

Приклад 3

Допустимо, що потрібно зареєструвати доменне ім'я "example.com" на 5 років. Обираємо реєстратора, який пропонує послугу реєстрації доменних імен за ціною 15 доларів США на рік. Таким чином, загальна вартість реєстрації доменного імені "example.com" складе 75 доларів США.

Приклад 4

Використовується сторонній DNS-сервер, який коштує 50 доларів США на місяць, але пропонує знижку 10% при щорічній оплаті. Таким чином, загальна вартість використання стороннього DNS-сервера для доменного імені "example.com" складе 6000 доларів США при щорічній оплаті.

Приклад 5

Припустимо, що ви хочете використовувати власний DNS-сервер, який ви придбали за 1000 доларів США. Ви також витратили 500 доларів США на його налаштування. Таким чином, загальна вартість використання власного DNS-сервера для доменного імені "example.com" складе 1500 доларів США.

Ці приклади лише ілюструють можливі варіанти розрахунків вартості реєстрації DNS-сервера. Конкретні витрати залежать від ваших потреб і вибору реєстратора та DNS-сервера.

3.6 Перевірка підключення до DNS-сервера

Перевірка	Команда	Результат	Значення
Підключення до DNS-сервера по IP-адресі	ping [IP-адреса DNS-сервера]	Reply from [IP-адреса DNS-сервера]: bytes=32 time=1ms TTL=128	Підключення успішне
Підключення до DNS-сервера по доменному імені	nslookup [доменне ім'я]	Server: [IP-адреса DNS-сервера] Name: [доменне ім'я] Address: [IP-адреса]	Підключення успішне

Рисунок 1.1 – Перевірка підключення до DNS-сервера

Перевірка підключення до DNS-сервера за допомогою команди ping

Для перевірки підключення до DNS-сервера за допомогою команди ping необхідно виконати наступні дії:

1. Відкрийте командний рядок.
2. Введіть наступну команду:

ping [IP-адреса DNS-сервера]

Наприклад, для перевірки підключення до DNS-сервера з IP-адресою 192.168.1.1, введіть наступну команду:

ping 192.168.1.1

Якщо команда ping успішно завершиться, ви побачите наступний результат:

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time=1ms TTL=128

Reply from 192.168.1.1: bytes=32 time=1ms TTL=128

Reply from 192.168.1.1: bytes=32 time=1ms TTL=128

Reply from 192.168.1.1: bytes=32 time=1ms TTL=128

--- 192.168.1.1 ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 2999ms

rtt min/avg/max/mdev = 1ms/1ms/1ms/0ms

Цей результат означає, що підключення до DNS-сервера успішне.

Перевірка підключення до DNS-сервера за допомогою команди nslookup

Для перевірки підключення до DNS-сервера за допомогою команди nslookup необхідно виконати наступні дії:

1. Відкрийте командний рядок.
2. Введіть наступну команду:

   nslookup [доменне ім'я]

Наприклад, для перевірки підключення до DNS-сервера для доменного імені "example.com", введіть наступну команду:

nslookup example.com

Якщо команда nslookup успішно завершиться, ви побачите наступний результат:

Server: 192.168.1.1

Address: 192.168.1.1#53

Name: example.com

Address: 192.168.1.10

Цей результат означає, що підключення до DNS-сервера успішне і DNS-сервер правильно відповідає на запит на визначення IP-адреси доменного імені "example.com".

Час виконання команди ping

Час виконання команди ping залежить від наступних факторів:

- Відстань між комп'ютером, з якого виконується команда, і DNS-сервером.
- Завантаження DNS-сервера.
- Завантаження мережі.

Зазвичай, час виконання команди ping становить від кількох мілісекунд до декількох секунд.

Час виконання команди nslookup

Час виконання команди nslookup залежить від наступних факторів:

- Відстань між комп'ютером, з якого виконується команда, і DNS-сервером.
- Завантаження DNS-сервера.
- Завантаження мережі.
- Складність запиту.

Зазвичай, час виконання команди nslookup становить від декількох мілісекунд до декількох секунд.

Висновки за розділом

Розрахунки параметрів DNS-сервера та запитів до них є важливим завданням, яке дозволяє забезпечити ефективну та надійну роботу DNS-системи.

Розрахунок навантаження на DNS-сервер дозволяє визначити, чи може сервер обробляти необхідну кількість запитів. Для цього необхідно врахувати такі фактори, як:

- Кількість запитів на хвилину
- Середній час на запит
- Ширина смуги пропускання

Якщо навантаження на DNS-сервер перевищує його пропускну здатність, то це може призвести до затримок або навіть відмов у роботі.

Розрахунок параметрів DNS-запиту дозволяє визначити, як швидко і ефективно буде оброблятися запит. Для цього необхідно врахувати такі фактори, як:

- Довжина доменного імені
- Тип запиту
- Метод розподілу навантаження

Якщо параметри запиту будуть занадто великими, то це може призвести до затримок у обробці.

На основі розрахунків параметрів DNS-сервера та запитів до них можна прийняти рішення про необхідність підвищення продуктивності DNS-системи. Для цього можна використовувати такі методи, як:

- Розподіл навантаження між декількома DNS-серверами
- Використання кешування

- Використання ефективних алгоритмів обробки запитів

Використання цих методів дозволяє забезпечити надійну та ефективну роботу DNS-системи навіть при високому навантаженні.

Ось кілька конкретних прикладів того, як можна використовувати розрахунки параметрів DNS-сервера та запитів до них:

- Якщо ви хочете створити DNS-сервер для невеликої мережі, то вам не потрібно проводити складні розрахунки. Однак, якщо ви плануєте використовувати DNS-сервер для великої мережі або для критичних до часу додатків, то вам необхідно провести ретельні розрахунки, щоб забезпечити надійну роботу DNS-системи.
- Якщо ви хочете оптимізувати роботу існуючого DNS-сервера, то ви можете провести розрахунки параметрів запитів, щоб визначити, які типи запитів обробляються повільно. Після цього ви можете внести зміни в конфігурацію DNS-сервера, щоб покращити обробку цих запитів.

Розрахунки параметрів DNS-сервера та запитів до них є важливим інструментом, який дозволяє забезпечити ефективну та надійну роботу DNS-системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кучернюк П. В. Основи теорії телекомунікацій: текст лекцій з дисципліни «Основи теорії телекомунікацій і радіотехніки. Київ: КПІ ім. Ігоря Сікорського, 2020. 290 с.

2. Кучернюк П.В. Комп'ютерні мережі: навчальний посібник з дисципліни «Комп'ютерні мережі та засоби телекомунікацій» для студентів спеціальності 7.05090201, 8.05090201 «Радіоелектронні апарати та засоби». Київ: НТУУ «КПІ», 2015 р. 238 с.

3. Кучернюк П.В. Технології моніторингу та трафік–інжинірингу в телекомунікаційних мережах: підручник для студ. спеціальності 172 «Телекомунікації та радіотехніка». Київ : КПІ ім. Ігоря Сікорського, 2021. 257 с.

4. Комп'ютерні мережі: підручник / [Азаров О. Д., Захарченко С. М., Кадук О. В. та ін.]. − Вінниця : ВНТУ, 2020. − 378 с.

5. Charanjeet S. How to Enable DNS Over HTTPS in Chrome, Firefox, Edge, Brave & More? Fossbytes. 2020.

6. Bumanglag K., Kettani H. On the Impact of DNS Over HTTPS Paradigm on Cyber Systems. 3rd International Conference on Information and Computer Technologies (ICICT), San Jose, CA, USA. 2020. P. 494-499.

7. Задерейко О.В., Трофименко О.Г., Прокоп Ю.В., Логінова Н.І., Кухаренко С.В. Аналіз витоків даних у інформаційних системах. Сучасна спеціальна техніка. 2021. № 3(66). С. 16-30. DOI: 10.36486/mst2411-3816.2021.3(66).2.

8. Основні поняття інформаційної безпеки [Електронний ресурс]-URL:https://press.vntu.edu.ua/index.php/vntu/catalog/download/604/1075/2185-1?inline=1.

9. Основи теорії телекомунікацій і радіотехніки. Частина 2

[Електронний ресурс]-URL:https://ela.kpi.ua./bitstream/123456789/48963/1/Osnovy_teorii_telekomunitaksii_radiotekhniky_2.pdf.

10. Основні поняття інформаційної безпеки, Азаров, 2020 [Електронний ресурс]-URL: http://pdf.lib.vntu.edu.ua/books/IRVC/Azarov_2020_378.pdf .

11. "DNS and BIND", Paul Mockapetris (видавництво "O'Reilly Media", 2015 рік).

12. "DNS-системи: принципи побудови і функціонування", І.О. Івасик, В.М. Міхеєнко, В.В. Рудь (видавництво "Навчальна книга - Богдан", 2012 рік).

13. "DNS-системи: основи та практика", В.А. Бойко, В.В. Рудь, В.М. Міхеєнко (видавництво "Навчальна книга - Богдан", 2014 рік).

14. "DNSSEC: The Future of DNS Security", Paul Mockapetris (видавництво "O'Reilly Media", 2019 рік).

15. "Комп'ютерні мережі", О.П. Сидоренко (видавництво "Навчальна книга - Богдан", 2022 рік).

16. "Інформаційна безпека", В.В. Рудь (видавництво "Навчальна книга - Богдан", 2022 рік).

17. "Комп'ютерні мережі: основи та практичні завдання", В.А. Бойко, В.В. Рудь, В.М. Міхеєнко (видавництво "Навчальна книга - Богдан", 2023 рік).

18. "Інформаційна безпека в хмарних технологіях", О.М. Якубенко (видавництво "Навчальна книга - Богдан", 2025 рік).

19. "Комп'ютерні мережі: основа сучасного світу", М.М. Пашкевич (видавництво "Вікар", 2022 рік).

20. "Безпека інформаційних систем", В.М. Міхеєнко (видавництво "Навчальна книга - Богдан", 2022 рік).

Додаток А

INVESTIGATION OF THE PRINCIPLES OF CONSTRUCTION AND FUNCTIONING OF DNS

1.1 Principles of DNS Protocol Construction

To establish correspondence between node names and their IP addresses, the Domain Name Service (DNS) protocol is employed. The domain name system constitutes a distributed database utilized by TCP/IP applications to establish this correspondence. DNS is also employed for email routing. The term "distributed database" implies that it is not stored on a single network node. Each node maintains its own information database and runs an application that sends queries to other servers. The DNS protocol enables communication between clients and servers.

A resolver provides access to DNS applications. A resolver is a subroutine used to create requests and interpret packets used by name servers on the network. For a network program, it is necessary to transform the node's name into an IP address before initiating a TCP connection or sending a datagram using UDP. The concepts of DNS are outlined in RFC 1034, while the details of development and DNS specifications are presented in RFC 1035.

1.1.1 Fundamentals of DNS

The DNS namespace possesses a tree-like hierarchical structure, as illustrated in Figure 1.1 depicting the organization of DNS.

Each node in Figure 1.1 carries a label with a length of 63 characters. The root of the tree is a special node without a label. Labels can consist of uppercase or lowercase letters. The domain name for any node in the tree is a sequence of labels starting from the root node, with labels separated by dots. Each node in the tree must have a unique domain name, but identical labels may be used at different points in the tree.

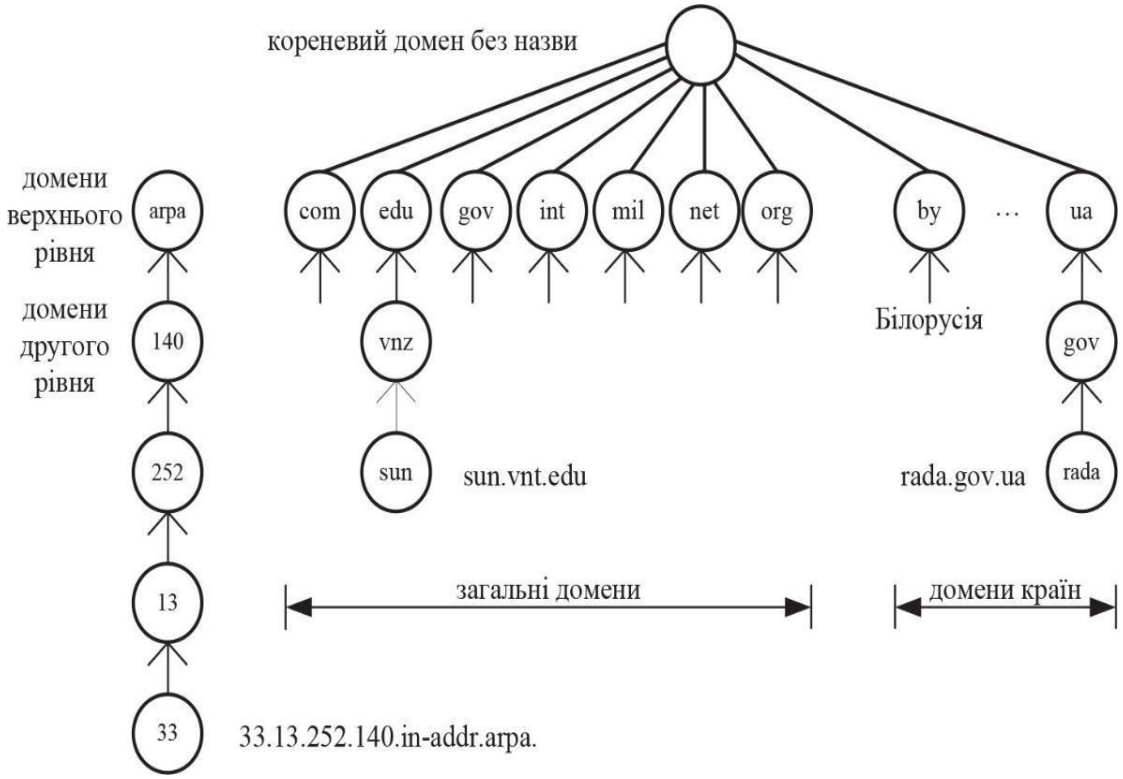


Figure 1.1 - Hierarchical Organization of DNS

An absolute domain name, denoted as an FQDN (fully qualified domain name), is a domain name ending with a dot, for example, "static.rada.gov.na." agra - a special domain used for mapping addresses to names.

Seven three-character domains are referred to as generic or organizational domains. Their classification is provided in Table 1.1. Names like .edu, .gov, .mil are reserved for U.S. organizations, while others can be used outside the U.S. In 2000, seven new top-level domains were introduced.

All two-character domains correspond to country codes, and their list can be found in ISO 3166. They are called country or geographical domains.

Table 1.1 - Generic Domains

Domain	Description	Domain (Introduced in 2000)	Description
.com	Commercial organizations	.qero	Aviation-related sectors
.edu	Educational organizations	.bis	Business-related entities
.gov	U.S. Government organizations	.coop	Non-commercial entities
.int	International organizations	.into	For unlimited use
.net	Networks	.museum	Museums
.org	Other organizations	Name	Individual persons
.mil	U.S. Military organizations	.pro	Accountants, lawyers, doctors

The delegation of responsibility within DNS is a crucial characteristic. No single organization serves the entire tree. Instead, a single organization, the NIC, serves only a portion of the tree (top-level domains), and responsibility for individual zones is delegated to other organizations. A zone is a separately managed part of the DNS tree. The organization responsible for managing a zone configures DNS servers (name servers) for that zone. When a new node appears in a zone (and, consequently, a new name), the zone administrator adds the name and IP address of the node to the DNS server's database.

A DNS server can handle one or more zones. The primary DNS server is created for a specific zone, along with one or more secondary servers. To ensure the stability of the DNS system in the event of a server failure, primary and secondary servers must be independent and redundant.

In the absence of necessary information, a DNS server contacts another DNS server. Each server must know how to contact the root DNS servers. In

turn, the root servers know the names and IP addresses of each official DNS server for all secondlevel domains.

Caching is a fundamental feature of DNS. When a DNS server receives information about the correspondence between a name and an address, it caches this information so that it can be used in case of subsequent requests without querying other servers.

1.1.2 DNS Message Format

The primary function of the DNS protocol is to manage the interaction between a DNS client and a DNS server. The DNS client sends a query, and the DNS server returns a response containing the necessary information for the client. The local DNS server forwards responses to clients and sends queries to other servers. Root servers only provide responses. For instance, if a program needs to determine the IP address for www.test.site.com, the client contacts the local DNS server, which, in turn, queries the root DNS server to find the IP address of the .com DNS server.

Next, the local DNS server sends a DNS query to the .com DNS server to discover the IP address of the site.com DNS server. After that, the local DNS server sends a query to the DNS server of the site.com zone. If the zone has subzones, an additional query for the test.site.com domain may be issued, providing the IP address for www.test.site.com in the response.

All DNS queries are categorized as recursive or iterative. A recursive query requires the DNS server receiving the query to perform the entire resolution process. For example, a resolver issues a recursive query to the local name server to translate a domain name into an IP address. In Figure 1.2, the resolver is activated through a system call at stage 1. The resolver then sends a DNS query to the local server (stage 2) and waits for a response (stage 9). The local DNS server carries out the resolver's query processing actions.

An iterative query requires the server in the response to provide the client with the IP address of the next DNS server in the hierarchy. Root servers only handle iterative queries. The local DNS server sends a query to the root

DNS server (stage 3) to obtain the name and IP address of the DNS server for the next level in the hierarchy (stage 4). This helps offload the root servers. The local DNS server may send queries to the next server in the hierarchy (stages 5, 6, 7, 8). Finally, the local server responds to the resolver (stage 9), and the resolver provides the IP address to the application (stage 10).

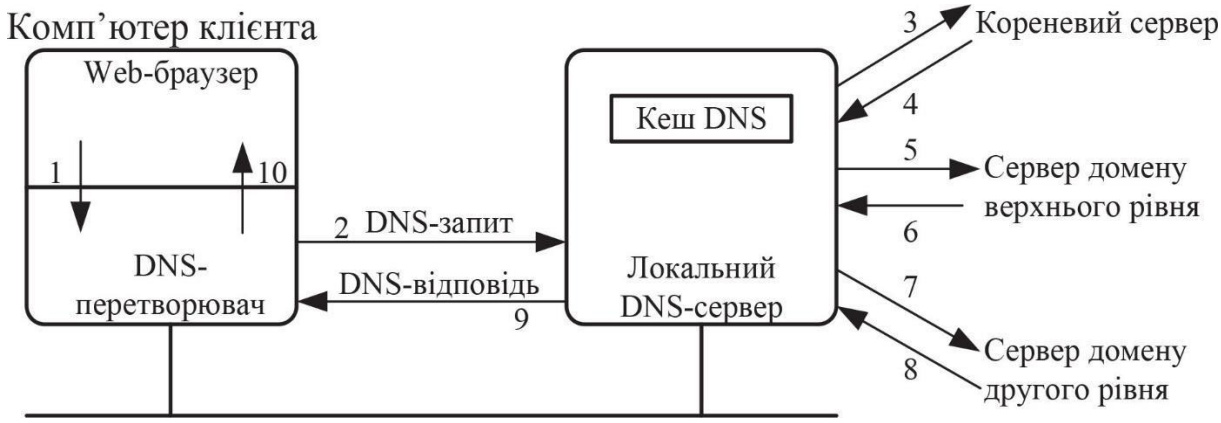


Figure 1.2 - Processing of DNS Query

However, the same format is used for both DNS queries and DNS responses.

Figure 1.3 illustrates the general format of a DNS message.

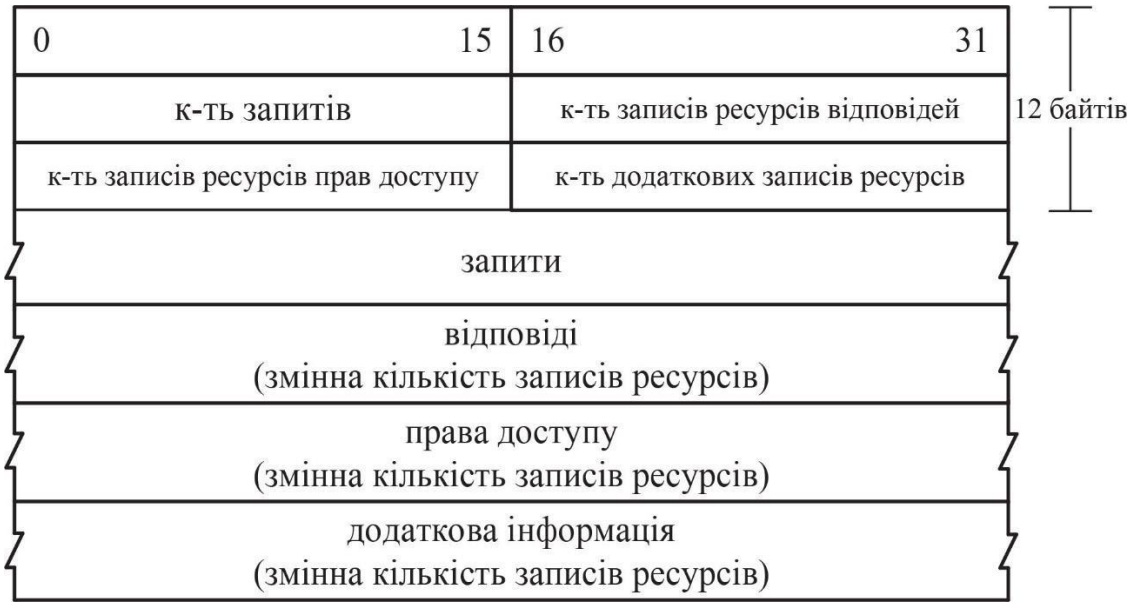


Figure 1.3 - DNS Message Format

The fixed 12-byte header contains the message, followed by four variablelength fields:

Identification field: Set by the client and returned by the server. This field allows the client to determine which query the response corresponds to.

16-bit Flags field: Divided into several parts, as shown in Figure 1.4.

1	4	1	1	1	1	3	4
QR	opcode	AA	TC	RD	RA	zero	rcode

Figure 1.4 - Flags Field of DNS Message Header

Here:

QR (Message Type): A 1-bit field; 0 indicates a query, and 1 indicates a response.

OP code (Operation Code): A 4-bit field. Typically contains the value 0 (standard query). Other values include 1 (inverse query) and server status query.

AA (Authoritative Answer): A 1-bit flag indicating an authoritative answer. The DNS server has authority for this domain in the query section.

TC (Truncated): A 1-bit field indicating truncation. For DNS, this means the full response size exceeds 512 bytes, but only the first 512 bytes of the response have been returned.

RD (Recursion Desired): A 1-bit field indicating the desire for recursion. This bit can be set in a query and then returned in a response. This flag requests the DNS server to process the query recursively, meaning the server should determine the required IP address itself rather than returning the address of another DNS server. If this bit is not set and the receiving DNS server does not have an authoritative answer, it returns a list of other DNS servers to contact for an answer. This is called an iterative query.

RA is a 1-bit field that means "recursion available". This bit is set to 1 in the response if the server supports recursion. Most DNS servers support recursion, with the exception of a few root servers that are too busy;

- zero is a 3-bit field that must be 0;

- rcode is a 4-bit response code field. Normal values: 0 (no errors) and 3 (name errors). A name error is returned only from a reputable DNS server and means that the domain name specified in the request does not exist;

The next four 16-bit fields indicate the number of items in the four variablelength fields that end the message. In the request, the number of questions is usually 1, and the other three counters are 0. In the returned query, the number of responses is at least 1, and the others can be either zero or non-zero. The format of each DNS query in the question field is shown in Figure 1.5 (usually there is only one question).

Figure 1.5 - Format of the DNS message field

Name:

- The query name is the name you are looking for. It looks like a sequence of one or more labels. Each label starts with a 1-byte counter that contains the number of bytes that follow it. The name ends with a byte equal to 0. It is a zero-length label and also the root label. Each byte counter must be in the range of 0 to 63, since the length of the label is limited to 63 bytes. This field may end at a trailing character that is not equal to 32 bytes. No placeholder is used. Figure 1.6 shows how the domain name is stored.

4	t	e	s	t	6	d	o	m	a	i	n	3	e	d	u	2	u	a	0

Figure 1.6 - Storing the domain name test. domanin. edu. ga in the part of the DNS query field name

- each question has a query type, and the answer has a type. There are about

20 different values, some of which are already outdated. Table 1.2 shows some of these values. The query type is a superset (a set of which this set is a subset) of the types: two of the values listed can only be used in questions. The most common type of record is type A, which means that the IP address for the specified name (query name) is required. A PTR query requires a name that matches the IP address.

Name	Numeric value	Description	Type	Query type
A	1	IP-address	Yes	Yes
NS	2	Server- DNS	Yes	Yes
CNAME	5	Canonical name	Yes	Yes
PTR	12	Pointer record	Yes	Yes
NINFO	13	Node information	Yes	Yes
MX	15	Mail exchange record	Yes	Yes
AXFR	252	Zone transfer request	No	Yes
ANY	255	Request all records	No	Yes

Table 1.2 - Values of the query type of the DNS message question field

- The query class is usually equal to 1, which indicates Internet addresses;

4) the last three fields of the DNS message are answers, authority and additional information. The general format is called an RR-resource record.

Figure 1.7 shows the general format of a resource record.

0 15	16 31
Domain name	
Type	Class
Lifetime	
Resource data length	
Course data	

Figure 1.7 - General format of a resource record

Let's analyze the corresponding fields in Figure 1.7:

- type indicates one of the types of RR codes. This is the same as the value of the request type. For Internet data, the class field is usually set to 1;
- domain name is the name to which the following resource data corresponds.

The name format is the same as shown in Figure 6.7 for the question field;

- the TTL-time-to-live field is the number of seconds during which the RR can be cached by the client. Usually RR is equal to 2 days;
- resource data length indicates the amount of resource data. For a type equal to 1 (record A), the resource data is a 4-byte IP address.

1.2 Principles of the DNS protocol

Both the local node and the centralized service can establish a correspondence between domain names (FQDN) and IP addresses [5]. The simplest solution is to manually create a text file named hosts. This file consists of a number of lines, each of which contains a record of the "IP address - domain name" type, for example, 77.47.129.30 - www.kpi.ua. The file must be placed in the appropriate system directory (for example, in the case of Windows, this is the \WINDOWS\system32\drivers\etc directory).

In general, DNS is used to solve the problem of matching domain names and IP addresses). The DNS service uses a client-server interaction scheme in its work. Within this interaction, DNS servers and DNS clients are defined. DNS servers maintain a distributed database of records, and DNS clients contact the servers with requests to determine the IP address for a particular domain name of a host or resource. Thus, DNS provides mechanisms for both naming hosts and looking up IP addresses of hosts by name.

The DNS service uses text files in much the same format as the hosts file, and these files can also be edited by the administrator.

Within this interaction, DNS servers and DNS clients are defined. DNS servers maintain a distributed database of records, and DNS clients query the servers to determine the IP address for a particular host or resource domain name. Thus, DNS provides mechanisms for both naming hosts and looking up IP addresses of hosts by name.

The DNS service uses text files in much the same format as the hosts file, and these files can also be manually edited by the administrator. However, the DNS service is based on a domain hierarchy, and each DNS server stores only a portion of the network names, not all of them, as is the case with hosts files. As the number of nodes in the network grows, the scaling problem is solved by creating new domains and subdomains of names and adding new servers to the DNS service.

A DNS server is created for each name domain. This server can store domain name-IP address records for the entire domain, including all its subdomains. More often, a domain server stores only names that end at the next lower level of the hierarchy compared to the domain name. It is with this organization of the DNS service that the load is distributed more or less evenly among all DNS servers in the network. For example, in the first case, the DNS server for the kpi.ua domain will store the mapping for all names ending in kpi.ua. In the second case, this server stores only names like fel.kpi.ua, fiot.kpi.ua, and all other records of lower hierarchy levels should be stored on the DNS servers of the fel and fiot subdomains.

It is necessary to properly configure DNS servers, zones and make the necessary records for the normal operation of DNS [5].

A DNS server is a computer with appropriate software applications, for example, the DNS server service in Windows, or the BIND service in UNIX systems. DNS servers maintain a DNS database with information about parts of the DNS domain tree structure and process name resolution requests from DNS clients. In response to a client's request, the DNS server provides the requested information, provides a link to another server that can respond to the

request, or reports that the information is unavailable or does not exist. DNS servers are divided into primary (authoritative) and backup (additional) servers.

For each zone, there can be only one primary server (on which you can make changes to zone information) and any number of backup servers (which receive zone information from the primary server). Installing redundant DNS servers allows you to solve two problems: to increase the reliability of the DNS system, to distribute client requests to different servers, thus increasing the performance of the DNS system.

A DNS zone is a single part of the namespace and IP addresses served by an authoritative server [5]. A server can serve several zones, and a zone can contain one or more Internet domains. For example, one server can be authorized for the kpi.ua and kpi.edu zones, each of which contains several domains. Adjacent domains, for example, kpi.ua, fel.kpi.ua and keoa.fel.kpi.ua can be transformed into separate zones by applying delegation, in which responsibility for a subdomain within the DNS namespace is assigned to a separate object (the corresponding DNS server).

Zone files contain records of zone resources in which the server is authoritative. Many DNS server implementations store zone data in text files; DNS servers on domain controllers running Windows 2000 or Windows Server 2003 can also store zone information in Active Directory.

There are two types of zones: forward and reverse lookup [5]. The first type maps FQDN names to IP addresses, while the second type maps IP addresses to full domain names. Thus, forward lookup zones serve queries to match FQDN names with IP addresses, and reverse lookup zones serve queries to match IP addresses with FQDN names.

If the name of the forward zone matches the domain name, the name of the reverse zone is formed from the network part of the IP address written in reverse order, to which the standard prefix in-addr.arpa is added. For example, the KEOA department is allocated the IP address block 10.12.80.0/24. The

name of the reverse zone is 80.12.10. in-addr.arpa. There are the concepts of the main zone, additional zone and stub zone (Fig. 1.8).

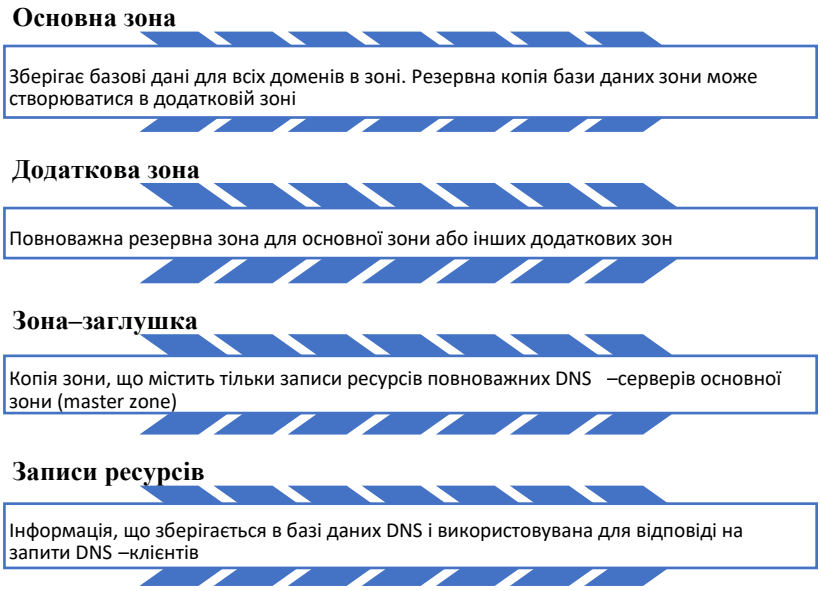


Figure 1.8 - DNS zones

Each DNS server contains the resource records that it needs to respond to queries related to its part of the DNS namespace. Resource records differ by type: for example, an address record (A), a canonical name (CNAME), a name server (NS), a mail exchanger (MX).

The most important types of DNS records are shown in Figure 1.9.

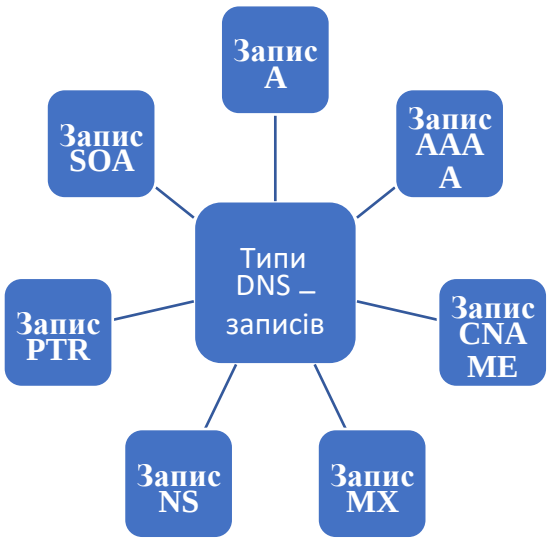


Figure 1.9 - Types of DNS records

A (address record) or address record - associates the name of a host with an IP address. For example, an A-record request for the name www.kpi.ua will return its IP address - 10.7.10.22.

The AAAA (IPv6 address record) associates a host name with an IPv6 address.

The CNAME record (canonical name record) is used to redirect to another name. It allows you to assign several names to one IP address.

MX (mail exchange) record or mail exchanger - specifies the mail exchange server(s) for this domain.

NS (name server) record - indicates the DNS server for this domain.

PTR (pointer) record - associates the IP address of a host with its canonical name. It is used in the reverse zone and is analogous to the A-record in the forward zone. Requesting a PTR record in the reverse zone in-addr.arpa to the IP address of a node will return the name (FQDN) of this node. In order to reduce the amount of unwanted correspondence (spam), many email recipient servers can check for a PTR record for the node from which the email is sent. In this case, the PTR record for the IP address must match the name of the sending mail server as it appears during the SMTP session.

SOA record (Start of Authority) or initial zone record - indicates the main DNS server of the zone, which stores the reference information about the zone, contains contact information of the person responsible for this zone, sets the parameters (serial number and time parameters) necessary for updating the zone information by the backup DNS servers.

SRV record (server selection) indicates the servers for some services; it is used, in particular, for Jabber and Active Directory.

TXT (Text) entry - a text entry used to make comments, notes, etc.

The named program that implements BIND, like any application layer service, uses TCP and UDP transport. If we can't access a particular computer, and we used to be able to, then the first thing to do is to check the availability

of the computer by its IP address. If this cannot be done, then we need to look for errors or malfunctions in the domain name service. The client part is the resolver name resolution procedure, and the server is the named program. A resolver is a set of procedures from the libc.a library system that allow an application program that is edited with these procedures to obtain a computer's IP address by a domain name, or a domain name by an IP address. These procedures call the resolver system component, which dialogs with the domain name server and thus serves the requests of user application programs. The general scheme of interaction between different BIND components can be seen in Figure 1.10.

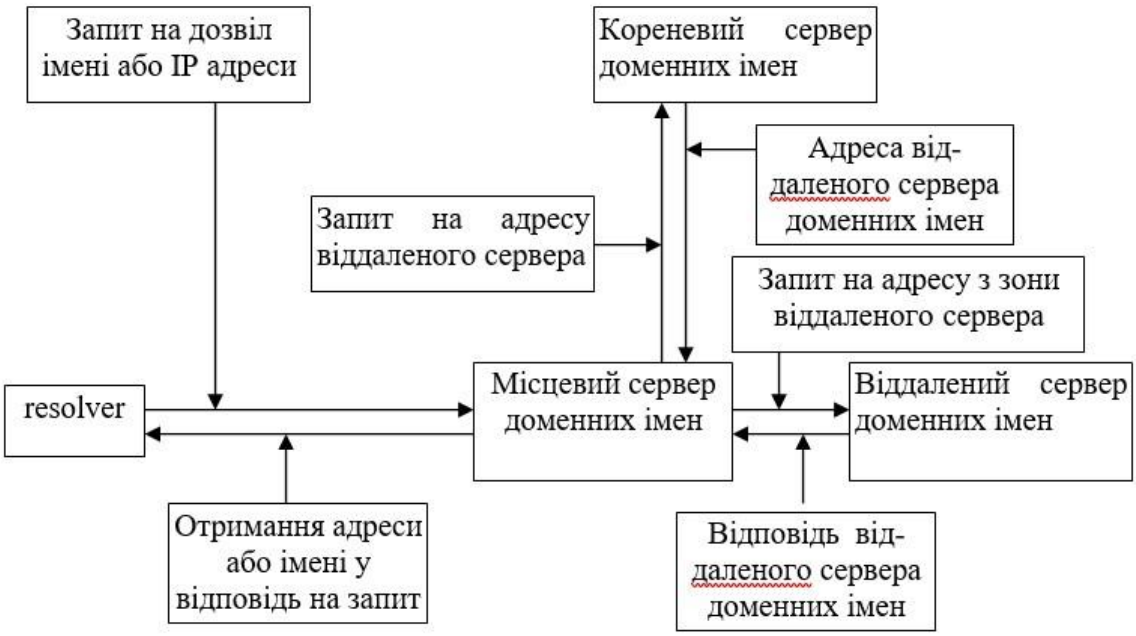


Figure 1.10 - Interaction of BIND components

A special database is managed by named, which consists of several files and contains correspondences between addresses and names, as well as the addresses of other BIND servers to which the data server can refer when searching for a name or address. Referring to Figure 1.10, let's look at two ways to resolve a request for an IP address by a domain name.

1st case. Request to obtain an IP address within the area of responsibility of a given local name server.

1) The application program requests the IP address by domain name from the local server through the resolver.

2) The local server informs the application program of the IP address of the requested name.

2nd case. An application program's request to a domain name server to obtain an IP address for a domain name from a domain that is in the area of responsibility of a remote domain name server, i.e. a server other than the one to which the computer making the request belongs.

In general, the scheme looks like this:

1) The application program contacts the local domain name server by IP address, informing it of the domain name.

2) The server determines that the address is not part of the domain and addresses the domain server, which requests the root domain name server.

3) The root domain name server informs the local domain name server of the server address of the requested domain.

4) The local domain name server queries the remote server for permission to authorize the request of its client, the application program.

5) The remote server reports the IP address to the local server.

The local server reports the IP address to the application program.

1.3 Problems with the DNS service

Let's recall the DNS algorithm for remotely looking up an IP address by name on the network:

- A host sends a DNS request to the IP address of the DNS server of its domain (it is set when configuring the IP protocol in the network OS), in which it specifies the name of the server whose IP address is to be found;

- The DNS server, upon receiving the request, looks through its name database to see if the name contained in the request is available. If the name is found, and therefore the corresponding IP address is found, the DNS server sends a DNS response to the requesting host, which contains the requested IP address. If the name specified in the request is not found by the DNS server in its name database, the DNS server sends the DNS request to one of the root DNS servers whose addresses are contained in the root. cache DNS server configuration file, and the procedure described in this paragraph is repeated until the name is found.

Analyzing the security vulnerability of this remote search scheme using the DNS protocol, we can draw a disappointing conclusion about the possibility of incorrect functioning of the DNS service, and we can identify the main causes of malfunctioning:

- remote attack - "False object of the DSS" (distributed computing system), i.e. introduction of an intermediate host through which the flow of information between the attacked object and the server will go or substitution (correction) of information about the zone;
- intersegment remote attack - an attack on DNS by falsifying the DNS server response;
- erroneous actions of the DNS server administrator, i.e. incorrect indication of the correspondence between the host IP address and its name.

Remote attacks on the DNS server. Practical searches and critical analysis of the security of the DNS service suggest that there are at least two possible variants of a remote attack using a false object on this service:

A "storm" of false DNS responses. The first variant of a remote attack targeting the DNS service is based on a variant of the typical remote attack "False RNS Object" [2]. In this case, the attacker continuously transmits a pre-prepared false DNS response on behalf of a real DNS server to the attacked host without receiving a DNS request (Fig. 1.11).

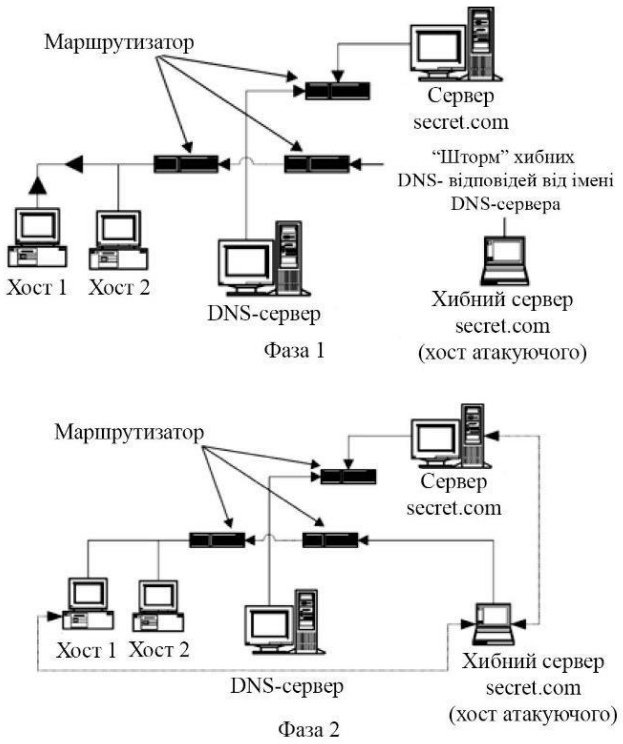


Figure 1.11 - A "storm" of false responses

In other words, the attacker creates a targeted "storm" of false DNS responses on the Internet. This is possible because the DNS query is usually transmitted using the UDP protocol, which does not have any means of identifying packets. The criteria that the host's network operating system imposes on the response received from the DNS server are, firstly, that the IP address of the sender of the response matches the IP address of the DNS server; secondly, the DNS response must contain the same name as the DNS query; thirdly, the DNS response must be sent to the same UDP port from which the DNS query was sent, and fourthly, the DNS response must contain the same value in the DNS header field (ID) as the transmitted DNS query.

Intercepting a DNS request. From the previously discussed scheme of remote DNS lookup, it follows that if the name specified in the request is not found by the

DNS server in its name database, the request is sent by the server to one of the root DNS servers, the addresses of which are contained in the root server configuration file. In other words, if the DNS server does not have information

about the host's request, it forwards the request further, which means that the DNS server itself is now the initiator of the remote DNS lookup. Therefore, nothing prevents an attacker from using the methods described above to transfer their attack directly to the DNS server [2]. In other words, the attacker will now target the DNS server, not the host, and false DNS responses will be sent by the attacker on behalf of the root DNS server to the attacked DNS server. It is important to take into account this feature of the DNS server. To speed up the work, each DNS server caches its own table of correspondence between host names and IP addresses in the memory area. The cache also contains dynamically changing information about the names and IP addresses of hosts found during the operation of the DNS server. That is, if a DNS server receives a request and does not find a corresponding entry in its cache table, it forwards the response to the next server and, upon receiving the response, enters the information found in the cache table. Thus, when the next request is received, the DNS server no longer needs to conduct a remote search, since the necessary information is already in memory.

From the analysis of the described remote DNS lookup scheme, it becomes obvious that if the attacker sends a false DNS response in response to a request from the DNS server (or in the case of a "storm" of false responses, he will constantly transmit them), then a corresponding record with false information will appear in the server's cache table, and subsequently all hosts that accessed this DNS server will be misinformed, and when accessing the host to which the attacker decided to change the route, communication with it will be carried out through the attacker's host using the "RVS object" scheme. And over time, this false information, which got into the DNS server cache, will spread to neighboring DNS servers of the highest levels.

Remote cross-segment attack on a DNS server. A much more general case is a cross-segment attack that does not require such strict conditions for its implementation, when the attacking and target DNS servers share a common physical transmission medium.

A cross-segment attack on a DNS server looks like this. Let's assume that the purpose of the attack is to "spoof" the IP address of the web server www.coolsite.com to the IP address of the server www.badsite.com for users of a certain subnet served by the DNS server ns.victim.com. In the first phase of the attack, ns.victim.com is provoked to search for information about the IP address www.coolsite.com by sending it a recursive query. In the second phase, the attacker sends a complex response to the ns.victim.com server on behalf of ns.coolsite.com, which is responsible for the coolsite.com domain. The false response contains the IP address www.badsite.com instead of the real IP address www.coolsite.com. The ns.victim.com server caches the information received, as a result of which, within a certain period of time (the value of this period is indicated in the TTL field of the false response and can be arbitrarily selected by the attacker), unsuspecting users are directed to www.badsite .com instead of the www.coolsite.com server (Figure 1.12).

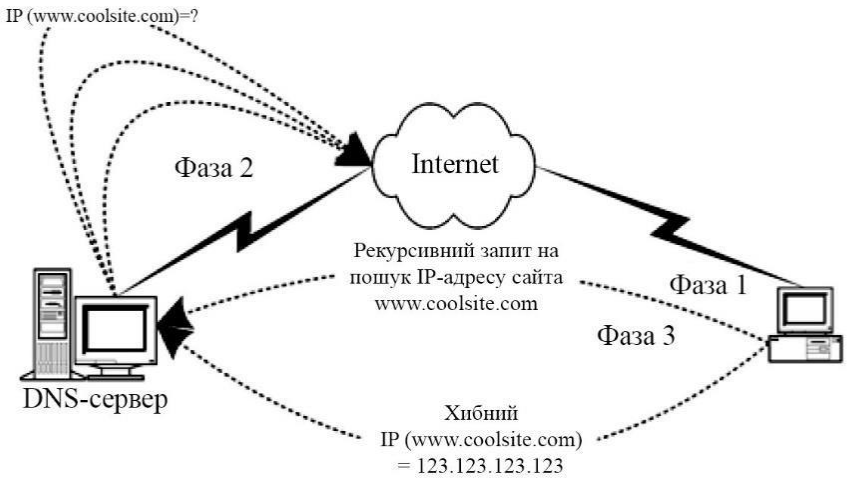


Figure 1.12 - Cross-segment remote attack

For a false response to be perceived as true by the ns.victim.com server, four conditions must be met:

- The IP address of the sender of the response must match the IP address of the requested server (in this case, ns.coolsite.com);

- The UDP port to which the response is sent must match the port from which the request was sent;
- the response identifier must match the request identifier;
- the response must contain the requested information (in this case, the IP address of the web server www.coolsite.com).

Obviously, fulfillment of the first and fourth conditions is not difficult for an attacker. The situation with the second and third conditions is much more complicated, since in the case of a cross-segment attack, the attacker has no way to intercept the original request and "spy" the necessary parameters.

Erroneous actions of the DNS server administrator. Erroneous actions of the DNS server administrator, i.e. incorrect specification of the correspondence between the host IP address and its name, can lead to the spread of the error to other DNS servers. Consequently, when you access the DNS server, it will give you the wrong IP address of the desired host.

As you can see from the section, there are a lot of problems in the network related to the correct functioning of the DNS service, and these are quite serious problems that can complicate the work of users and network administrators. The next section will discuss some modern solutions and tips for avoiding these problems.

Some solutions to DNS service problems. The best solution from the security point of view is to refuse to use the DNS service at all in the protected segment. Of course, it will be very inconvenient for users to completely abandon the use of names when addressing hosts. Therefore, you can offer the following compromise solution: use names, but abandon the remote DNS lookup mechanism that was used before the advent of the DNS service with dedicated DNS servers. Back then, each machine on the network had a hosts file that contained information about the corresponding names and IP addresses of all hosts on the network. Obviously, today an administrator can enter information about only the most frequently visited network servers by users of a given segment into such a file.

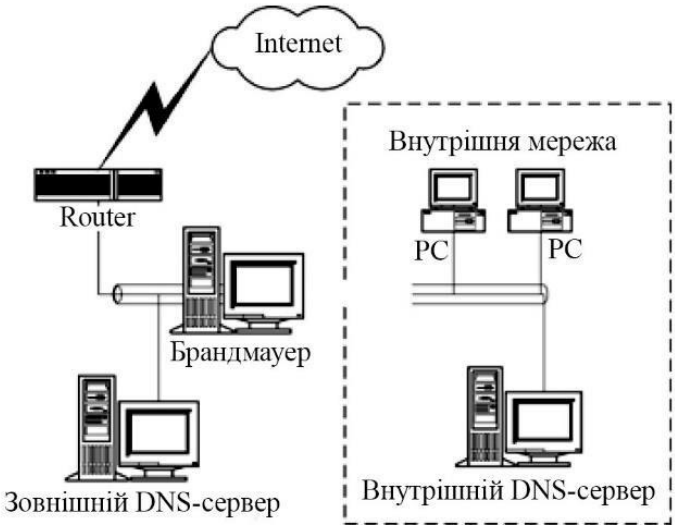


Figure 1.13 - An internal DNS server serves the corporate network and is not visible from the outside. An external DNS server provides only part of the network information

To make a remote attack more difficult, you can suggest that administrators use the TCP protocol for the DNS service instead of the default UDP protocol (although it is not obvious from the documentation how to change it). This will make it much more difficult for an attacker to send a false DNS response to a host without receiving a DNS request.

We can also suggest using the BIND (Berkley Internet Name Daemon) application suite. Starting with version 4.9.3, several directives and DNS record types have been added to the BIND specification to improve the security of name servers. The xfrnets directive of the initial boot file (/etc/named.boot) allows you to specify a list of IP addresses of networks and name servers to which a given server is authorized to forward information within a zone (zone forwarding operation). The second important innovation is the introduction of a special type of TXT resource record called SECURE_ZONE. This record manages the list of machines and networks (by IP address) that can query this name server. But despite these innovations, a number of other measures need to be taken to prevent DNS spoofing attacks.

Among them, the most common is the installation of two DNS servers: external and internal (see Figure 1.8).

The internal DNS server is designed exclusively to serve internal network clients. It stores all information about the hosts of the corporate network. Due to the use of records of the SECURE_ZONE type, this server can be requested only by internal hosts. Moreover, a filter is installed on the firewall that does not allow IP packets sent to the corporate network and destined for port 53 of the UDP and TCP protocols of the internal DNS server. That is, such a DNS server is made invisible from the outside. However, it can access the DNS servers of the Internet for information.

The latest version of BIND 8.2.2. includes support (RFC 2065) for cryptographic digital signatures, i.e. it is no longer a standard DNS protocol, but an extended one, in which the body of the DNS request will include a digital signature. This solution will almost completely secure the work with the DNS service. Unfortunately, the desired result can only be achieved through the large-scale implementation of new protocols, which is associated with significant organizational difficulties and cannot be implemented in a short time.

Conclusions to the chapter

DNS is a system that allows users to enter human-readable website names in the search bar and for the browser to receive the IP address of the resource to be accessed. The system for storing DNS data (zones) is distributed, but the servers that store the data are organized hierarchically. At the top level are the root DNS servers, below them are the DNS servers of geographic zones, and even lower are the local DNS servers. A DNS query made by a browser first refers to local DNS servers (resolve), and then they look for the required DNS record in the hierarchy above. The DNS data of each domain is called DNS zones and stores DNS records of different types. DNS zones can be managed using paid and free services.

Додаток Б

УДК 621.396

О.С. Жученко, д.т.н., доцент,
Я.Д. Васєв, магістрант
Національний університет «Полтавська політехніка імені Юрія Кондратюка»

ОСНОВИ ПОБУДОВИ ТА ФУНКЦІОНУВАННЯ DNS

Для отримання відповідності між іменами вузлів та їх IP-адресами використовується протокол DNS (Domain Name Service). Система доменних імен являє собою розподілену базу даних, що використовуються застосуванням TCP/IP для встановлення даної відповідальності. Також DNS використовується для маршрутизації електронної пошти. На верхньому рівні знаходяться кореневі DNS-сервери, нижче - DNS сервери географічних зон, ще на нижчий рівень - локальні DNS-сервери.

Взаємодією між DNS-клієнтом та DNS-сервером керує протокол DNS. DNS-клієнт відправляє запит, а DNS-сервер повертає відповідь, що містить необхідну для клієнта інформацію. DNS-запит може бути рекурсивним або ітеративним. На рис.1 показано, що на етапі 1 перетворювач активізується через системний виклик. Далі перетворювач надсилає DNS-запит локальному серверу (етап 2) і чекає відповіді (етап 9).

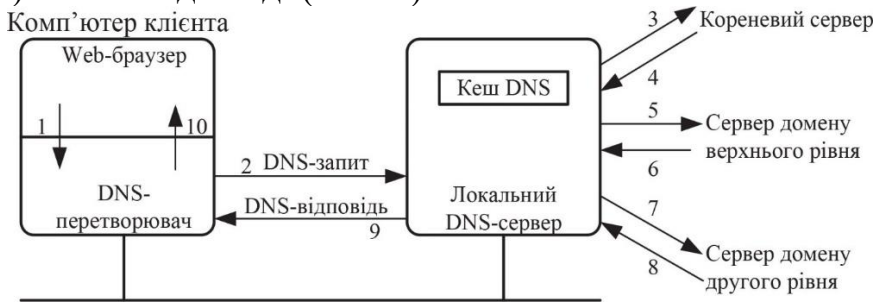


Рис.1. Опрацювання DNS-запиту

Відповідність між доменними іменами (FQDN) і IP–адресами може встановлюватися як засобами локального вузла, так і засобами централізованої служби. Існують поняття основної зони, додаткової зони і зони – заглушки (рис. 2).

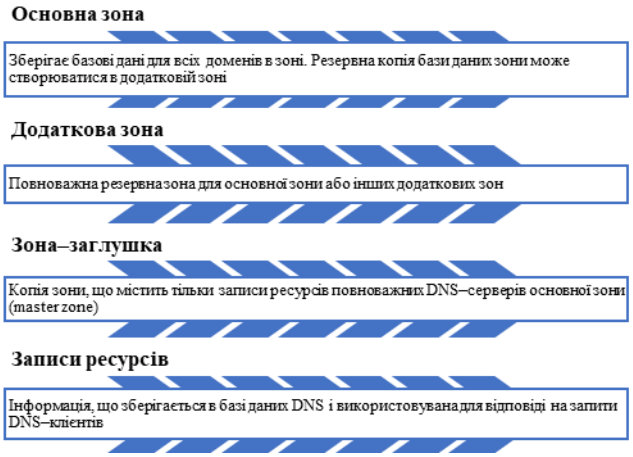


Рис. 2. Зони DNS

Кожен DNS – сервер містить записи ресурсів, необхідні йому для відповіді на запити, що відносяться до його частини простору імен DNS. Записи ресурсів розрізняються за типами: наприклад, адресний запис (A), канонічне ім'я (CNAME), сервер імен (NS), поштовий обмінник (MX).

Найбільш важливі типи DNS–записів (рис. 3).

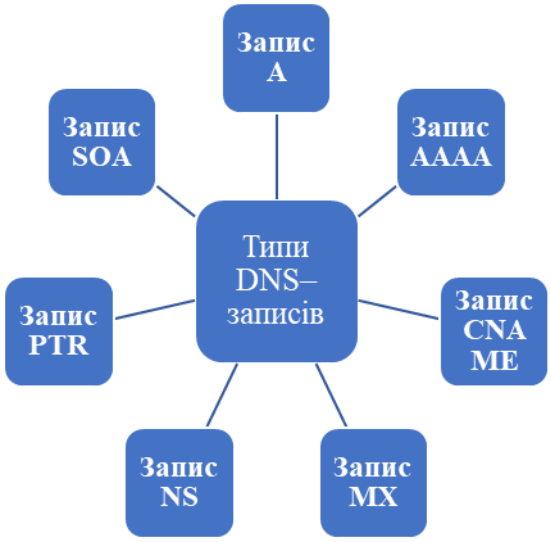


Рис. 3. Типи DNS–записів

Найбільш популярна програма підтримки доменної адресації є named. Вона реалізує BIND. В системі Windows NT 4.0 є свій сервер доменних імен, який підтримує специфікацію BIND. Це сервер доменних імен який був запропонований в університеті Берклі. Він забезпечує пошук IP адрес та доменних імен для довільного вузла мережі.

Можна виділити основні причини неправильного функціонування DNS-служби: віддалені атаки на DNS-сервер, перехоплення запиту DNS, помилкові дії адміністратора DNS-сервера.

ЛІТЕРАТУРА:

1. Комп'ютерні мережі: підручник / [Азаров О. Д., Захарченко С. М., Кадук О. В. та ін.]. − Вінниця : ВНТУ, 2020. − 378 с.

2. Кучернюк П. В. Основи теорії телекомунікацій: текст лекцій з дисципліни «Основи теорії телекомунікацій і радіотехніки. Київ: КПІ ім. Ігоря Сікорського, 2020. 290 с.

3. Кучернюк П.В. Комп'ютерні мережі: навчальний посібник з дисципліни «Комп'ютерні мережі та засоби телекомунікацій» для студентів спеціальності 7.05090201, 8.05090201 «Радіоелектронні апарати та засоби». Київ: НТУУ «КПІ», 2015 р. 238 с.

BASICS OF BUILDING AND FUNCTIONING OF DNS

***O.S. Zhuchenko**, Doctor of Science. Associate Professor,*
***Y.D. Vasiev**, Master's student*
National University «Yuri Kondratyuk Poltava Polytechnic»

Додаток В

Міністерство освіти та науки України
Національний університет «Полтавська політехніка імені Юрія Кондратюка»

Кафедра автоматики, електроніки та телекомунікацій

Дослідження принципів функціонування системи доменних імен

Кваліфікаційна робота магістра

Виконав:
Студент 601дТТ групи Васєв Я.Д.

Керівник:
канд. техн. наук, доцент Жученко О.С.

Полтава 2024

Актуальність роботи полягає у тому, що система доменних імен є дуже актуальною темою в сучасному інформаційному суспільстві, відіграє ключову роль в Інтернеті, забезпечуючи перетворення легкозапам'ятовуваних доменних імен на IP-адреси, що дозволяє комп'ютерам знаходити один одного в мережі. Дослідження в цій області може принести користь для розуміння та покращення ефективності та безпеки інтернет-інфраструктури. Також, з урахуванням постійного розвитку технологій та збільшення кількості проблем безпеки в Інтернеті, вивчення системи доменних імен залишається актуальним завданням.

Метою роботи є розробка рекомендацій для покращення ефективності, безпеки та інновацій в цій важливій складовій інтернет-інфраструктури.

Для виконання поставленої мети в роботи необхідно виконати наступні **завдання:**

- розібратися у принципах побудови та функціонування DNS;
- дослідити роботу сервера створеного на віртуальній машині;
- розрахувати параметри серверу та запитів до нього.

Об’єкт дослідження – функціонування, значення та загальне використання DNS.

Предмет дослідження – система доменних імен, DNS-сервер, запити та повідомлення.

Принципи побудови протоколу DNS

Для отримання відповідності між іменами вузлів та їх IP - адресами використовується протокол DNS (Domain Name Service). Система доменних імен являє собою розподілену базу даних, що використовуються застосуванням TCP/IP для встановлення даної відповідальності. Також DNS використовується для маршрутизації електронної пошти.

Протокол DNS дозволяє клієнтам і серверам спілкуватися між собою.

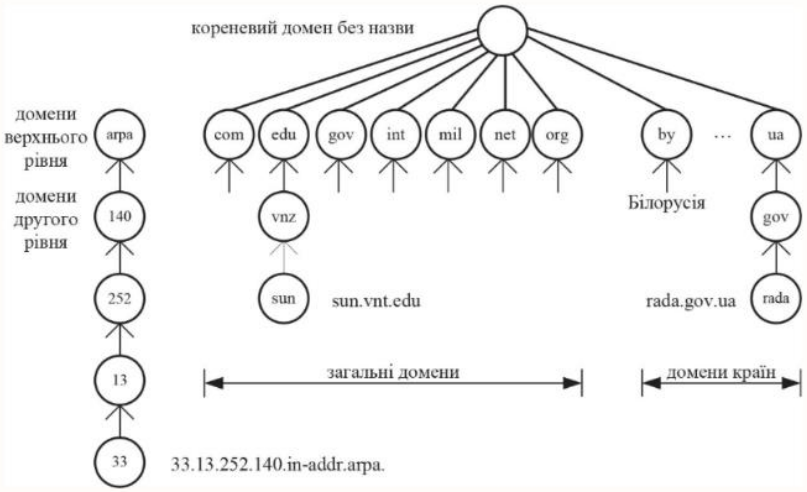
Визначник (resolver) виконує доступ до DNS застосування. Визначник– це підпрограма, що використовується для створення відправлення й інтерпретації пакетів, що використовуються серверами імен у мережі.

Основи DNS

Існують три типи доменів:

1) arpa – це спеціальний домен, що використовується для зіставлення адрес та імен;
2) загальні або організаційні домени — їх усього сім, мають у собі трисимвольне значення (com, edu, gov, int, mil, net, org);
3) домени країн або географічні — мають двосимвольне значення (ua, ru, ag, by).

Усі вони утворюють разом одне "абсолютне доменне ім'я".



Формат повідомлення DNS

Головною функцією протоколу DNS є керування взаємодією між DNS-клієнтом та DNS-сервером. DNS-клієнт відправляє запит, а DNS-сервер повертає відповідь, що містить необхідну для клієнта інформацію. Локальний DNS-сервер надсилає відповіді клієнтам і видає запити іншим серверам. Кореневі сервери надають лише відповіді.

Всі DNS – запити поділяються на рекурсивні або ітеративні.
Рекурсивний запит вимагає, щоб DNS-сервер, який приймає запит, сам виконував перетворення. Ітеративний запит вимагає, щоб даний сервер у відповіді клієнту надав IP-адресу наступного в ієрархії DNS-сервера. Кореневі сервери обслуговують лише ітеративні запити.

Принципи функціонування DNS

Розподілена база даних

DNS використовує розподілену базу даних, що дозволяє швидку обробку запитів та збереження резервних копій.

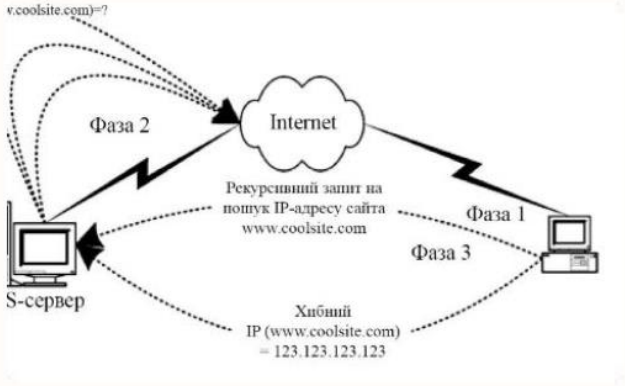
Кешування

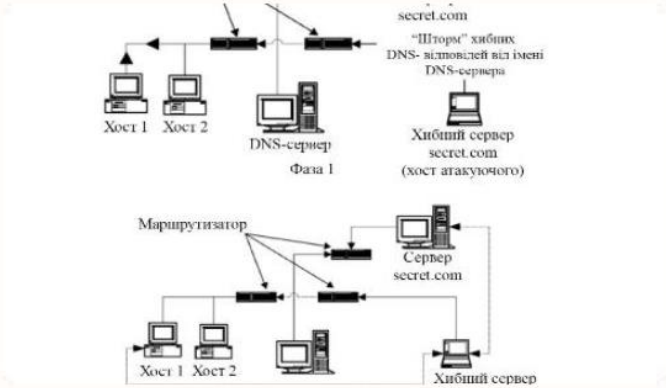
Сервери DNS використовують кеш для збереження результатів попередніх запитів, що зменшує час відповіді на майбутні запити.

Загальнодоступність

DNS є загальнодоступним сервісом, що дозволяє веб-браузерам та іншим програмам звертатися до серверів DNS для отримання необхідної інформації про доменні імена.

Проблеми функціонування DNS-служби





Однією з основних проблем є атаки на DNS, такі як DNS-підміна та DNS-отруєння. Ці атаки спрямовані на перехоплення або модифікацію DNS-запитів, що може призвести до перенаправлення користувачів на фальшиві веб-сайти або обмеження доступу до легітимних ресурсів.

Другою проблемою є скарги на відсутність стандартизації та сумісності між різними реалізаціями DNS. Це може призводити до того, що деякі ресурси можуть бути недоступні для користувачів, які використовують певні DNS-клієнти чи сервери.

Також важливою є проблема витоку конфіденційної інформації через DNS-запити. Нешифрована передача інформації може стати об'єктом атак та порушити приватність користувачів.

Значення мережевих аналізаторів та їх особливості

Мережеві аналізатори, також відомі як аналізатори трафіку, протоколів або пакетів, є програмами або пристроями, призначеними для моніторингу та аналізу трафіку, що проходить через мережу. Ці аналізатори можуть опрацьовувати необроблені двійкові дані, перетворюючи їх у зручний для читання формат і сприяючи подальшому аналізу мережі.

Компоненти мережевих аналізаторів, такі як декодер, буфери, фільтри захоплення та обладнання, виконують такі важливі функції:

- збір (перший етап, мережеві аналізатори встановлюють мережеві картки інтерфейсу (NIC) в "безладний режим");
- перетворення (другий етап, перетворення зібраних необроблених двійкових даних в зручний для читання формат);
- аналіз (третій етап, протоколи, які були зібрані на попередньому етапі та використовуються у мережевому трафіку, можуть бути оглянуті та проаналізовані для отримання інформації).

Загальне використання мережевих аналізаторів

У законному використанні вони можуть приносити безліч переваг мережі. Адміністратори мережі активно використовують ці засоби для оптимізації мережі, виявлення та усунення проблем, а також для захисту від порушень та затримок.

Водночас вони можуть використовуватися для здійснення незаконних дій, таких як крадіжка конфіденційної інформації чи підслуховування мережі хакерами.

Wireshark

Wireshark, відомий аналізатор мережевих пакетів, представляє собою відкритий код для аналізу усунення несправностей, моніторингу та протоколів мережі, доступний безкоштовно.

Також варто відзначити, що Wireshark дозволяє експортувати дані пакетів у різні формати файлів. Зручний графічний інтерфейс користувача (GUI) робить аналіз пакетів легким та доступним, що робить цей аналізатор популярним. Крім того, програма також підтримує використання через інтерфейс командного рядка.

```
root@kali:~# tcpdump -i eth0 icmp -c 10 -w file.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
10 packets captured
10 packets received by filter
0 packets dropped by kernel
root@kali:~# tcpdump -r file.pcap
reading from file file.pcap, link-type EN10MB (Ethernet)
13:03:24.157259 IP 192.168.0.7 > 104.28.6.89: ICMP echo request, id 2153, seq 2836,
13:03:24.313600 IP 104.28.6.89 > 192.168.0.7: ICMP echo reply, id 2153, seq 2836, l
13:03:25.160549 IP 192.168.0.7 > 104.28.6.89: ICMP echo request, id 2153, seq 2837,
13:03:25.314956 IP 104.28.6.89 > 192.168.0.7: ICMP echo reply, id 2153, seq 2837, l
13:03:26.163092 IP 192.168.0.7 > 104.28.6.89: ICMP echo request, id 2153, seq 2838,
13:03:26.317409 IP 104.28.6.89 > 192.168.0.7: ICMP echo reply, id 2153, seq 2838, l
13:03:27.165332 IP 192.168.0.7 > 104.28.6.89: ICMP echo request, id 2153, seq 2839,
13:03:27.319381 IP 104.28.6.89 > 192.168.0.7: ICMP echo reply, id 2153, seq 2839, l
13:03:28.168736 IP 192.168.0.7 > 104.28.6.89: ICMP echo request, id 2153, seq 2840,
13:03:28.323582 IP 104.28.6.89 > 192.168.0.7: ICMP echo reply, id 2153, seq 2840, l
```

Tcpdump

Tcpdump - це ще один визнаний мережевий аналізатор, який працює у текстовому інтерфейсі командного рядка. Його можливості включають перегляд ідентифікаторів мережі, аналіз поведінки мережі, перегляд паролів, веб-сайтів та їхнього вмісту, який відвідує користувач.

Проте для використання tcpdump користувачеві потрібні права адміністратора. Важливо відзначити, що tcpdump не такий інтуїтивний та простий у використанні, як Wireshark. Для звичайного користувача може бути важко розібратися з командним рядком, оскільки він опосередковано взаємодіє з обмеженою кількістю протоколів.

Висновок

- DNS - це система, яка дозволяє користувачам вводити зрозумілі для людини назви сайтів у пошуковий рядок, а браузеру отримувати IP-адресу ресурсу, до якого потрібно звернутися.
- Одним із ключових висновків є те, що мережеві аналізатори забезпечують операторам мережі високий рівень видимості в процесах передачі даних, дозволяючи вчасно виявляти і вирішувати проблеми з підтримкою стабільності та продуктивності мережі.
- З'ясовано, що процес обміну повідомлень у DNS ґрунтується на чітко визначених протоколах та стандартах, зокрема на принципах роботи протоколу DNS, який визначає взаємодію між різними складовими системи.
- Мережеві аналізатори, як ключовий інструмент у сфері інформаційних технологій, виявляються невід'ємною частиною адміністрування та моніторингу комп'ютерних мереж.
- На основі розрахунків параметрів DNS-сервера та запитів до них можна прийняти рішення про необхідність підвищення продуктивності DNS-системи. Для цього можна використовувати такі методи, як:
- Розподіл навантаження між декількома DNS-серверами
- Використання кешування
- Використання ефективних алгоритмів обробки запитів

Використання цих методів дозволяє забезпечити надійну та ефективну роботу DNS-системи навіть при високому навантаженні.